



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Herramientas de ayuda al seguimiento del desarrollo de
Trabajos Fin de Estudios

Autor/es

ÁLVARO JOSÉ GARCÍA PÉREZ

Director/es

JUAN JOSÉ OLARTE LARREA y FRANCISCO JOSÉ GARCÍA IZQUIERDO ,

Facultad

Escuela de Máster y Doctorado de la Universidad de La Rioja

Titulación

Máster Universitario en Tecnologías Informáticas

Departamento

MATEMÁTICAS Y COMPUTACIÓN

Curso académico

2018-19



Herramientas de ayuda al seguimiento del desarrollo de Trabajos Fin de Estudios, de ÁLVARO JOSÉ GARCÍA PÉREZ

(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.

Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los titulares del copyright.

© El autor, 2019

© Universidad de La Rioja, 2019

publicaciones.unirioja.es

E-mail: publicaciones@unirioja.es

Trabajo de Fin de Máster

Herramientas de ayuda al seguimiento del desarrollo de Trabajos Fin de Estudios

Autor

Álvaro José García Pérez

Tutores:

Juan José Olarte Larrea
Francisco José García Izquierdo

MÁSTER:

Máster en Tecnologías Informáticas (853M)

Escuela de Máster y Doctorado



**UNIVERSIDAD
DE LA RIOJA**

AÑO ACADÉMICO: 2018/2019

Resumen

El Trabajo de Fin de Máster consistirá en realizar un proyecto de reingeniería sobre la aplicación ya existente para el seguimiento del desarrollo de los Trabajos Fin de Grado.

El primer paso consistirá en llevar a cabo una investigación a través de la cual se detecten las mejoras necesarias en el funcionamiento y la usabilidad de la aplicación.

Una vez analizado el pequeño estudio estadístico, se procederá a implementar la nueva aplicación utilizando el framework Symfony, así como todos los vendor que sean necesarios.

Abstract

This Master Final Project will deal with the realization of a reengineering project on the already existing application for the monitoring of the development of the Final Degree Projects.

The first step will consist of carrying out an investigation through which necessary improvements in its operation and usability of the application will be identified.

Once the small statistical study has been analysed, the new application will be implemented using the Symfony framework, as well as all the necessary vendors.

Memoria

1.	DOP	1
1.1	Antecedentes	1
1.2	Objetivos	1
1.2.1	Objetivos académicos	1
1.3	Integrantes	2
1.4	Decisiones y justificación	2
1.4.1	Tecnologías	2
1.4.2	Metodología	3
1.5	Riesgos	4
1.5.1	Riesgos específicos del proyecto	4
1.5.2	Riesgos generales de cualquier proyecto	5
1.6	Entregables	6
1.7	Alcance	6
1.8	Planificación	6
2.	ANÁLISIS	10
2.1	Tutores: resultado de las entrevistas	10
2.2	Alumnos: resultado del cuestionario	11
2.3	Funcionalidad de la futura aplicación	12
2.4	Actores	15
2.5	Requisitos no funcionales	16
2.6	Requisitos funcionales	17
3.	DISEÑO DEL SISTEMA	21
3.1	Diagrama de clases	21
3.2	Diseño de datos	23
3.3	Requisitos de datos	25
3.4	Diseño de interfaces	26
3.4.1	Interfaz del menú de la aplicación del director	27

3.4.2	Interfaz del menú de la aplicación del alumno.....	28
3.4.3	Acceso a la aplicación.....	29
3.4.4	Recuperación de contraseñas.....	29
3.4.5	Alta de alumnos o directores.....	31
3.4.6	Alta/Modificación de datos del TFG.....	31
3.4.7	Añadir/Modificar representante de la empresa.....	32
3.4.8	Búsqueda de TFG.....	32
3.4.9	Añadir/Modificar replanificación.....	33
3.4.10	Añadir/Modificar progreso actual del TFG.....	33
3.4.11	Repositorio.....	34
3.4.12	Modificar opciones de aviso al correo electrónico.....	35
3.4.13	Selección de alumnos.....	36
3.4.14	Envío de notificaciones.....	36
3.4.15	Gráficos.....	37
4.	IMPLEMENTACIÓN.....	41
4.1	Introducción.....	41
4.2	Tecnologías empleadas.....	41
4.2.1	Doctrine.....	42
4.2.2	Charts.js.....	44
4.2.3	Twig.....	45
4.2.4	jQuery.....	46
4.2.5	Swiftmailer.....	46
4.2.6	Sonata.....	47
4.3	Seguridad.....	48
4.3.1	Control de acceso.....	48
4.3.1	Gestión de contraseñas.....	49
5.	SEGUIMIENTO Y CONTROL.....	50
6.	CONCLUSIÓN.....	51
6.1	Lecciones aprendidas.....	51
6.2	Conclusiones de valoración personal.....	52
7.	REFERENCIAS BIBLIOGRÁFICAS.....	53
8.	AGRADECIMIENTOS.....	54
	Anexo I: Plantilla para la entrevista con los directores de TFG.....	1

Anexo II: Plantilla realizada para el cuestionario de exalumnos.....	1
Anexo III: Instalación y puesta en marcha del entorno de trabajo.....	1
Anexo IV: Plantilla de ayuda para el desarrollo de Symfony.....	1
Anexo V: Casos de uso.....	1

Figuras

FIGURA: HORAS ESTIMADAS PARA CADA TAREA A REALIZAR	7
FIGURA: DIAGRAMA DE GANTT.....	8
FIGURA: ACTORES.....	15
FIGURA: MODELO MVC	21
FIGURA: DIAGRAMA DE CLASES I	22
FIGURA: DIAGRAMA DE CLASES II	22
FIGURA: BASE DE DATOS	24
FIGURA: INTERFAZ – MENÚ DE LA APLICACIÓN DEL DIRECTOR	27
FIGURA: INTERFAZ – MENÚ DE LA APLICACIÓN DEL ALUMNO.....	28
FIGURA: INTERFAZ – VALIDACIÓN DE USUARIOS.....	29
FIGURA: INTERFAZ – RECUPERACIÓN DE CONTRASEÑA I.....	29
FIGURA: INTERFAZ – RECUPERACIÓN DE CONTRASEÑA II.....	30
FIGURA: INTERFAZ – RECUPERACIÓN DE CONTRASEÑA III	30
FIGURA: INTERFAZ – ALTA DE ALUMNO.....	31
FIGURA: INTERFAZ – MODIFICACIÓN DATOS DEL TFG.....	31
FIGURA: INTERFAZ – MODIFICAR DATOS RESPONSABLES DE LA EMPRESA	32
FIGURA: INTERFAZ – BÚSQUEDA DE PROYECTOS POR PERFIL.....	32
FIGURA: INTERFAZ – AÑADIR PLAN DE SEGUIMIENTO	33
FIGURA: INTERFAZ – AÑADIR PUNTO DE SEGUIMIENTO	33
FIGURA: INTERFAZ – REPOSITORIO I	34
FIGURA: INTERFAZ – REPOSITORIO II	34
FIGURA: INTERFAZ – REPOSITORIO III	35
FIGURA: INTERFAZ – OPCIONES DE AVISO	35
FIGURA: INTERFAZ – SELECCIÓN DE ALUMNOS	36
FIGURA: INTERFAZ – ENVÍO DE NOTIFICACIONES.....	36
FIGURA: INTERFAZ – GRÁFICO COMPARATIVO DEL PROGRESO VS. PLANIFICACIÓN....	37
FIGURA: INTERFAZ – GRÁFICOS DE EVOLUCIÓN DEL PROGRESO (LÍNEAS)	38
FIGURA: INTERFAZ – GRÁFICOS DE EVOLUCIÓN DEL PROGRESO (BARRAS).....	39
FIGURA: INTERFAZ – GRÁFICOS DE PUNTOS.....	40
FIGURA: COMPARATIVA DE LIBRERÍAS JS PARA GENERAR GRÁFICAS	44

FIGURA: HORAS REALES PARA CADA TAREA A REALIZAR	50
FIGURA: VERSIONES DE SYMFONY ACTUALMENTE	2
FIGURA: DIAGRAMA DE PAQUETES - APLICACIÓN.....	1
FIGURA: DIAGRAMA DE CASOS DE USO - SEGURIDAD.....	1
FIGURA: DIAGRAMA DE CASOS DE USO - GESTIÓN DE PERSONAS.....	4
FIGURA: DIAGRAMA DE CASOS DE USO - GESTIÓN DE PLANIFICACIÓN	5
FIGURA: DIAGRAMA DE CASOS DE USO - GESTIÓN DE PROGRESO	6
FIGURA: DIAGRAMA DE CASOS DE USO - GESTIÓN DE TFGS	8
FIGURA: DIAGRAMA DE CASOS DE USO - GESTIÓN DE GRÁFICAS	10
FIGURA: DIAGRAMA DE CASOS DE USO - GESTIÓN DE OPCIONES DE AVISO	11
FIGURA: DIAGRAMA DE CASOS DE USO - GESTIÓN DE REPOSITORIO	12

1. DOP

1.1 Antecedentes

Actualmente se dispone de una aplicación para ayudar en el seguimiento de los Trabajos Fin de Grado (en adelante TFG). Dicha aplicación consta de los siguientes módulos:

- Gestión de datos de alumnos y profesores.
- Gestión de la temporalización del proyecto.
- Gestión del perfil de los diferentes TFG.
- Gestión de entregables.
- Gestión de reuniones.
- Gestión de redes sociales.

Dicha aplicación ha sido desarrollada con las siguientes tecnologías: PHP, MYSQL, JavaScript y Mailer.

1.2 Objetivos

El objetivo de este Trabajo Fin de Máster (en adelante TFM) es realizar una reingeniería de la aplicación existente.

Para ello se deberá recopilar información acerca de la aplicación, buscando los motivos por los cuales tanto profesores como alumnos no la utilizan todo lo que sería deseable, y aquellas características que debería tener la aplicación para utilizarla por su parte.

1.2.1 *Objetivos académicos*

Aprovechar la realización del TFM para **actualizar mis conocimientos** sobre:

- Gestión de proyectos.

- Lenguajes, entornos de desarrollo integrados (IDEs) y Framework.
- Utilización de bases de datos mediante ORMs.

1.3 Integrantes

Francisco José García y Juan José Olarte: Tutores del TFM en la UR.

Comunicación con mis tutores.

- Uso de correo electrónico para solucionar dudas sencillas.
- Reuniones presenciales en caso de ser necesario.

1.4 Decisiones y justificación

1.4.1 Tecnologías

Se acordó con los tutores utilizar alguna de las tecnologías vistas durante el máster. De esta manera, se decidió utilizar Symfony como framework de php, así como la base de datos MariaDB. Para la capa de presentación se decidió utilizar Twig así como el framework Bootstrap, acompañados de HTML, CSS y JavaScript.

Por ello, se decidió utilizar el framework Symfony 3, ya que es uno de los framework más robustos y obliga a aplicar buenas prácticas de programación. Además, es apto para desarrollos de gran tamaño y es muy modular. Por el contrario, Symfony tiene la fama merecida de ser muy complejo; por ello, el proyecto contemplará más tiempo de documentación y formación y menos de desarrollo que el que podría llevarse a cabo en un proyecto normal.

Se utilizará como ORM Doctrine, ya que su principal ventaja es una integración totalmente transparente con el framework, lo que puede simplificar mucho las tareas a realizar, aunque tiene una serie de inconvenientes que podría abocar al fracaso de este TFM:

- Mala documentación.
- Poca actividad de desarrollo.
- Su excesiva complejidad.
- Dificultad para encajarlo en los proyectos.

Se utilizará Bootstrap para generar la capa de presentación mediante este framework que nos permite crear interfaces web a base de CSS y JavaScript.

1.4.2 Metodología

Se tiene experiencia en proyectos similares ya que se han realizado con anterioridad varios trabajos fin de carrera, así como una experiencia de 6 años como programador de aplicaciones web. Aunque puede ser un hándicap el programar en php y la utilización del framework Symfony. Esto puede suponer un riesgo que produzca bloqueos y retrasos en la fecha final de entrega del TFM.

Por eso, se ha optado por un modelo iterativo-incremental. Para minimizar la incertidumbre, se combinará con el desarrollo de prototipos que ayuden a comprender mejor el framework.

Se realizará un prototipo que tendrá como fin familiarizarse con el framework. Será un mero ejemplo con la tabla personas; al segundo prototipo se le añadirá la gestión de roles.

La primera iteración contendrá los requisitos de back-end: gestión de personas, roles y tablas maestras. Y una primera versión de la interfaz del back-end.

En la segunda iteración, se desarrollará la gestión de todos los datos referentes al TFG, así como todas las tablas auxiliares que sean necesarias.

En la tercera iteración, se desarrollará toda la gestión del control de seguimiento de los alumnos: replanificaciones, seguimiento y gráficas comparativas.

En la cuarta iteración, se fijará la interfaz de back-end, se realizarán pruebas exhaustivas del correcto funcionamiento de la aplicación y se implementará el envío de mensajes de los tutores a los alumnos, ya sea correspondencia inmediata como mensajería programada.

1.5 Riesgos

1.5.1 Riesgos específicos del proyecto

Uso de Symfony y Doctrine

Symfony es uno de los framework más exigentes para los principiantes. Existen riesgos de bloqueos y retrasos durante la fase de desarrollo, por ello se dedicarán una serie de horas para la búsqueda de documentación y realizar cursos de formación. Estas horas serán paralelas al diseño de la aplicación.

Recolección de requisitos de la aplicación

Si no se realiza correctamente la preparación del estudio, puede desembocar en que dicho estudio no tenga ninguna utilidad. Para minimizar este riesgo, se realizará una encuesta previa que deberá realizarse personalmente con los profesores que conforman el Departamento de Matemáticas y Computación de la Universidad de la Rioja, dentro del área de informática. Tras esta serie de entrevistas, se recopilará la información recogida y se elaborará una encuesta con los ítems obtenidos de este primer paso. Una vez que los exalumnos hayan completado las encuestas, será el momento de procesar los resultados y llegar a una serie de conclusiones que nos den información acerca del poco uso de la aplicación que se encuentra en funcionamiento y la funcionalidad que esperan tanto profesores como alumnos en la utilización de esta nueva aplicación.

1.5.2 Riesgos generales de cualquier proyecto

Pérdida de información

Para proteger los programas, licencias y configuraciones del equipo de desarrollo se usará virtualización. Las características del sistema son: Sistema Operativo Windows 10 con phpStorm, Apache, PHP, MariaDB y Symfony; todos configurados e interrelacionados. El tamaño de la máquina es de 12GB. Se realizará una copia semanal de la máquina virtual completa (se guardan la primera y las dos últimas copias en un disco duro externo). Por otro lado, se realizará una copia diaria de seguridad del directorio donde se encuentra el proyecto. Se almacenará en el equipo anfitrión y cada dos días se guardará en el otro disco duro externo. Cada tres días se subirá una copia a Google Drive.

Enfermedad o accidente por parte del proyectante en el desarrollo del proyecto

Si el proyectante sufre una enfermedad o un accidente que le impide continuar con el normal desarrollo del proyecto. En este caso, se procederá con el reajuste de la planificación inicial, lo cual llevará a un retraso de todas las acciones que resten para concluir dicho proyecto.

Cambios en las especificaciones del programa

Si en el transcurso del proyecto surgen nuevos requisitos que no han sido recogidos en el análisis inicial, se realizará un estudio del impacto que sufrirá en consecuencia el proyecto. En caso de realizar los cambios, se tendrá que estudiar el alcance final del proyecto.

Estimaciones mal realizadas

La experiencia es un factor importante en el desarrollo de un proyecto de estas características. Aun con la experiencia de haber realizado varios proyectos, no se puede asegurar que la estimación sea la correcta; se puede intuir que si todo es correcto, puede variar en un +- 10%.

Requerimiento de algún programa con licencia

En un principio, todo el software necesario para la elaboración del proyecto será de distribución libre y sin costes para su uso. En el caso de detectarse algún

software cuyo uso requiera una licencia de pago, se estudiará el pago de la correspondiente licencia o evaluar posibles alternativas.

Ausencia del tutor académico de la UR

En principio, al contar con dos directores se minimiza este riesgo, aunque es posible que pueda darse el problema de que un profesor esté de baja y el otro tenga una ausencia corta, por ejemplo, un congreso. En este caso, se aplazaría la reunión y se proseguiría con las siguientes tareas del proyecto. Si se diera la casuística de que los dos tutores se encontrasen de baja prolongada, se buscaría el consejo del director del máster.

1.6 Entregables

- Documento de Objetivos del Proyecto.
- Documento de análisis de requisitos.
- Documento de diseño UML.
- Documento de diseño de la base de datos.
- Memoria.
- Listado de pruebas.
- Manual de usuario e instalación.
- Presentación.

1.7 Alcance

El alcance final del proyecto corresponderá con el análisis, diseño, implementación y pruebas de las cuatro iteraciones de las que se hacer referencia en el punto 1.4.2 Metodologías.

1.8 Planificación

Debido a los retrasos por diferentes problemas surgidos durante el transcurso del TFM se procede a realizar una replanificación con fecha 6 de mayo del 2019.

En esta replanificación, se debe tener en cuenta el trabajo realizado con anterioridad. Las tareas ya realizadas han sido las siguientes:

- Se dedicaron 40 horas para diseñar el guion, realizar las entrevistas personales a los profesores, crear un cuestionario para exalumnos a partir de los datos obtenidos y examinar dichos datos.
- Se dedicaron 20 horas a la formación en Symfony3, Doctrine y Swift Mailer.
- Se dedicaron 40 horas para el diseño de la base de datos, y a partir de la base de datos obtener las entidades y comenzar a trabajar con ellas.
- Se dedicaron 8 horas para el seguimiento del TFM.

Las horas totales estimadas para cada una de las tareas son las siguientes:

Nombre de la tarea	Duración horas
Preparación	20
Análisis	50
Documentación	55
Diseño	40
Desarrollo	170
Pruebas	20
Seguimiento	14
Total	369

Figura: Horas estimadas para cada tarea a realizar

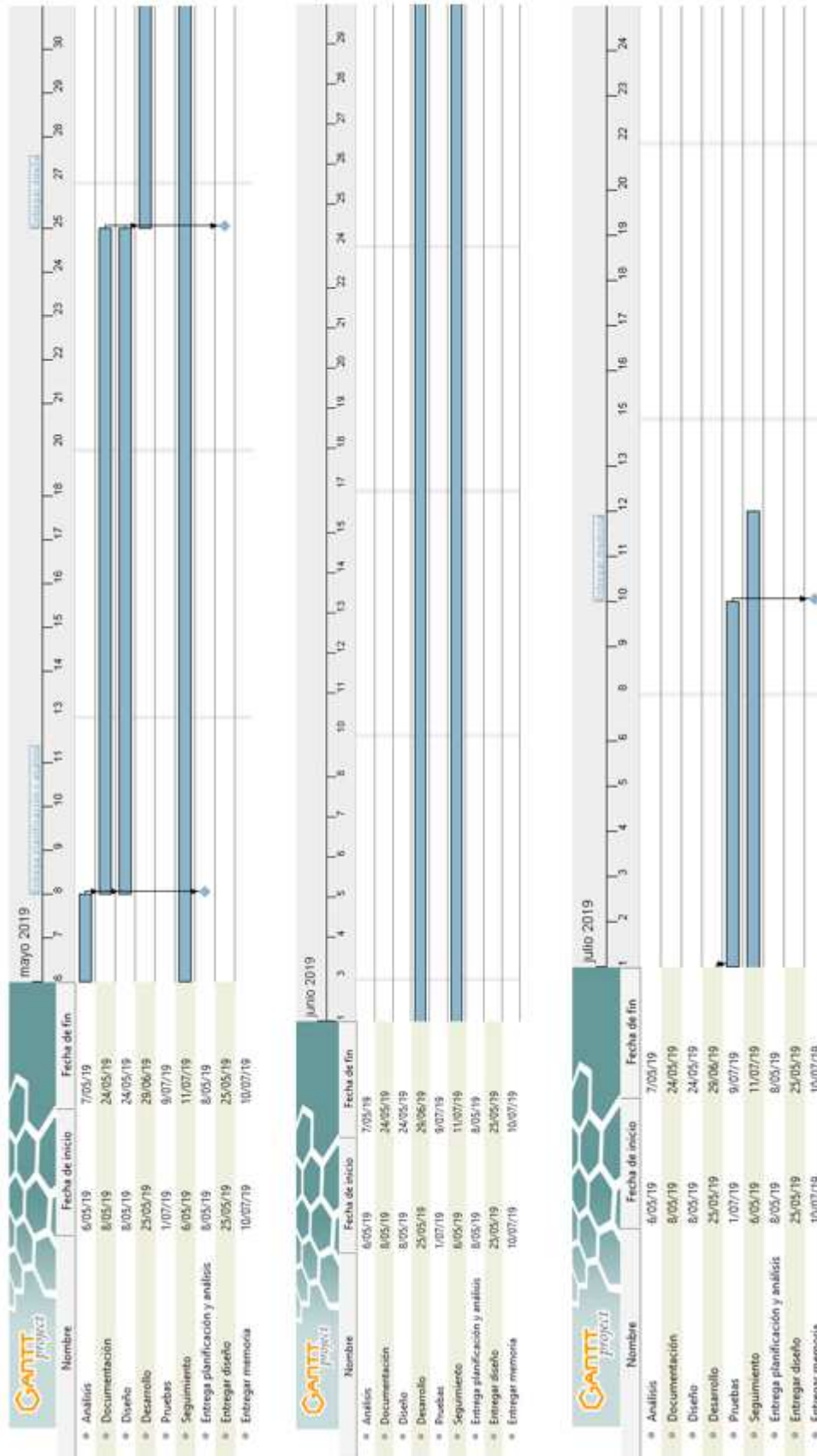


Figura: Diagrama de Gantt



Se establece un horario de 5 horas diarias de lunes a viernes y de 6 horas los sábados. Se dejan los domingos para recuperar horas ante problemas que pudiesen surgir durante la semana.

2. ANÁLISIS

Este TFM consiste en realizar un proyecto de reingeniería sobre la aplicación ya existente para el seguimiento del desarrollo de los Trabajos Fin de Grado.

Por ello, es importante que el proceso de investigación se realice correctamente para poder detectar la funcionalidad y usabilidad que los usuarios de la aplicación requieren, y proceder a su posterior implementación.

En primer lugar, se hace un estudio superficial de la aplicación actual, viendo las opciones y funcionamiento que ofrece tanto a los tutores del TFG como a los alumnos.

En segundo lugar, a partir de los requisitos recogidos en la antigua aplicación, se diseña un guion para realizar las entrevistas con los profesores del departamento que han dirigido algún TFG en cursos anteriores. Se puede consultar el guion realizado para las entrevistas con los tutores en el Anexo I.

En tercer lugar, se analizaron los resultados obtenidos a través de las entrevistas personales con los tutores. Y con estos resultados se creó la plantilla para el cuestionario que sería lanzado a los exalumnos a través del grupo cerrado creado en Facebook. Es importante destacar que en este tipo de cuestionarios las preguntas deben ser lo más cerradas posible, ya que las preguntas abiertas, por norma general, no suelen dar buen resultado en el estudio. Se puede consultar el cuestionario lanzado a los alumnos en el Anexo II.

En cuarto y último lugar, se recogen todas las opiniones vertidas tanto por tutores como alumnos y se señalan los requisitos que debería tener la aplicación.

2.1 Tutores: resultado de las entrevistas

Participaron en el cuestionario 11 profesores, de los cuales 2 no habían utilizado apenas la aplicación y 5 de ellos la habían utilizado esporádicamente.

Los resultados de este estudio son los siguientes:

Con respecto a la aplicación actual.

- Mejorar el control de versiones añadiendo feedback entre el tutor y el alumno.
- Cambiar el tipo de menú actual por uno más usable y atractivo.
- Integrar la aplicación con blackboard.
- Mejorar la usabilidad de la aplicación.
- Simplicidad de la aplicación.
- La aplicación debe ser más proactiva (recordar tareas o búsqueda de proyectos parecidos).

Mejoras que podrían implementarse en una nueva aplicación.

- Recibir en el correo electrónico las actualizaciones y poder acceder a ellas sin tener que validarse en el sistema.
- Integración con los TFG ubicados en la biblioteca.
- Interactuar con la aplicación de depósito del TFG.
- Integrar el repositorio con Office 365 y poder interactuar en el mismo documento.
- Autogenerar actas en la propia aplicación.
- Crear un asistente de documentos (resumen, conclusiones...).
- Que el tribunal pueda ver el progreso del alumno.
- Compartir documentación (plantillas de documentos, buenas prácticas, TFG de otros compañeros, fechas relevantes...).

2.2 Alumnos: resultado del cuestionario

Se puede acceder a los resultados del cuestionario en la siguiente dirección URL:

<https://docs.google.com/forms/d/1shLn5tkaiqLWVPopdWVNJD5M9HaVDbtiZXtPMWSPB4M/viewanalytics>.

El cuestionario ha sido respondido por 16 alumnos. Quisiera hacer un pequeño inciso y agradecerles su tiempo empleado para rellenar dicho

cuestionario y ayudarme a la hora de recabar información, la cual me facilitó hallar los requisitos funcionales.

Tras este pequeño estudio estadístico de los resultados, se puede extraer la siguiente información:

- La aplicación es poco amigable y, a la vez, poco usable. Esto provoca que los alumnos no hagan uso de ella.
- También se ha destacado que las gráficas comparativas no son muy reales, ya que hay alumnos que comienzan utilizando la aplicación y posteriormente dejan de utilizarla.
- Otro punto importante es que actualmente la aplicación no permite el intercambio de mensajes entre el tutor y el alumno de los documentos subidos.

A continuación, se mostrarán por orden de preferencia una lista con la funcionalidad que les hubiese gustado a los exalumnos que tuviera la aplicación:

- Notificación de eventos (fechas de publicación de tribunales, actos...).
- Sistema para poder asignar tutorías a los alumnos.
- Entorno para resolver dudas en el cual pudieran intervenir alumnos y profesores.
- Mensajería entre el tutor y alumno, chat o video conferencia.
- Generador de actas a través de un formulario.
- Plantillas de los diferentes documentos, así como documentos de buenas prácticas.
- Notificaciones al teléfono móvil mediante SMS.
- Conectar la aplicación con los proyectos que están accesibles en la biblioteca

2.3 Funcionalidad de la futura aplicación

La nueva aplicación debe tener la funcionalidad que ya poseía la anterior aplicación:

- Sistema de registro de usuarios
 - Estandarización del registro.
 - Proceso de recuperación de contraseña.
- Sistema de gestión de planificación y seguimiento de los TFM
 - Añadir replanificaciones del TFG.
 - Actualizar progreso del TFG.
 - Acceso a la información de planificación y seguimiento por parte del profesor.
- Generación de estadísticas
 - Evolución de la planificación.
 - Evolución del progreso.
 - Datos comparativos entre planificación y seguimiento de los proyectos propios.
 - Datos comparativos del estado de los proyectos o entre el alumno y la media de los demás alumnos.
- Gestión de los datos del TFM
 - Configuración del perfil del TFM.
 - Acceso a datos de directores de cada TFM.
- Gestión de repositorio de documentos
 - Subida de documentos por parte del alumno y del profesor.
- Gestión de correos electrónicos
 - Avisos a los implicados en diferentes acciones (subida de documentos, cambio de planificación, añadir nuevo punto de control del TFM...).
- Gestión de redes sociales
 - Publicar mensajes en diferentes redes sociales.

Dentro de la funcionalidad de la aplicación se ejecutarán las siguientes mejoras:

- En el caso de pérdida de la contraseña, la persona afectada restablecerá su contraseña después de realizar el proceso de autenticación correspondiente. Introduciendo el DNI y el correo electrónico, le llegará un enlace al correo electrónico con un token para restablecer la contraseña.

De esta manera, tendrá el control sobre la contraseña y si alguna persona introduce sus datos en la regeneración de contraseña, esta no será instantánea y siempre requerirá que el usuario confirme el cambio.

- En el caso de que introduzca datos erróneos en la planificación o en la evolución del TFG, podrá corregir los datos antes de que termine el día en el que realizó la modificación.
- Se centralizará el envío de documentos entre el tutor y el alumno en un repositorio; además, se incorpora la función de añadir comentarios a cada entrega realizada. La interfaz será tipo foro.
- Se añade la posibilidad de descargar todos los documentos de un alumno en un fichero zip.
- Para cada archivo se añadirá un documento Json con la información de dicho documento y añadiendo todos los comentarios que se hagan al mismo.
- Se dará la posibilidad al tutor de gestionar todos los documentos de todos los alumnos, sin tener que seleccionar a ninguno previamente y realizar las opciones de subida al repositorio, ver información y comentar.
- Se añaden en el progreso del TFG 4 campos para guardar las horas dedicadas en cada fase y poder mostrar unas gráficas más detalladas.
- Se añadirá una opción a través de la cual tanto alumnos como tutores puedan definir qué alertas desean recibir en su correo electrónico.
- Las tablas maestras podrán ser modificadas desde el propio programa a través de sonata (parámetros, áreas, tipos de documento, tecnologías y tipos de proyecto).

Implementación de nuevas funcionalidades.

- El tutor podrá enviar correos electrónicos a los alumnos desde la aplicación, pudiendo adjuntar documentos si es necesario. El envío de correos puede ser inmediato o diferido. Además, se podrá programar que un correo se mande cada X días a unos determinados alumnos.
- Se añadirá un acceso directo a los alumnos para que puedan consultar los TFG depositados en la biblioteca de la Universidad de la Rioja.

2.4 Actores

Analizando los componentes involucrados en el proyecto, se obtienen los siguientes actores:

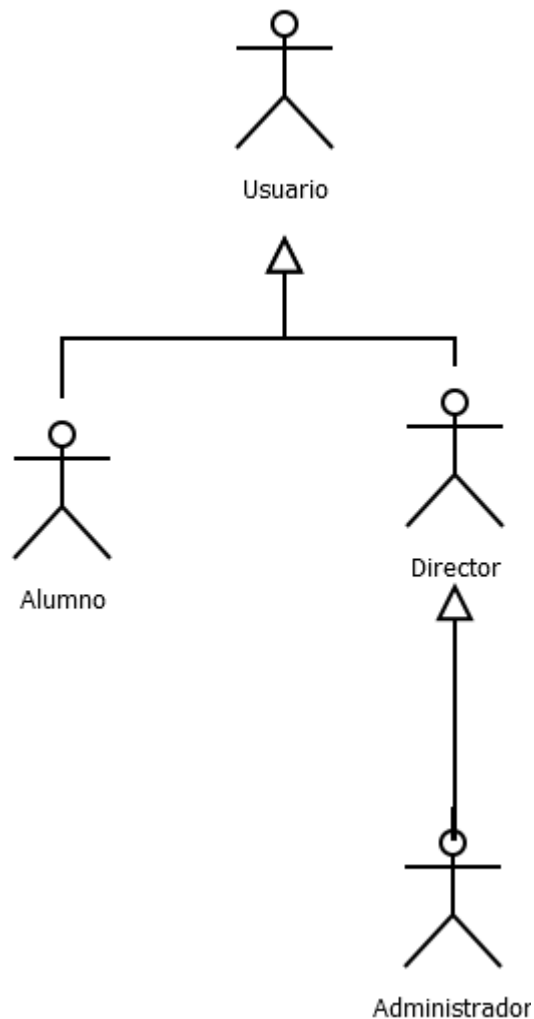


Figura: Actores

Usuario. Viene impuesto por Symfony para gestionar la seguridad de la aplicación.

Alumno. Es la persona que está realizando el TFG y que tendrá el mayor peso en la aplicación ya que será el encargado de hacer sus propias planificaciones, progresos, subir documentos...

Director. Supervisará el trabajo de su tutorando. Podrá realizar un seguimiento del estado de este mediante las gráficas comparativas y descargarse los documentos que los alumnos suban al repositorio. También realizará modificaciones y podrá subirlas, además de realizar comentarios sobre dichos documentos.

Administrador. En principio, podrá ser cualquier director; en caso de querer limitarlo, habría que añadir el role en la tabla persona y modificar el archivo de seguridad para que solo ese nuevo rol tuviera acceso a las actividades de administración.

2.5 Requisitos no funcionales

- **General**

G1 El programa no realizará por sí mismo ninguna operación sin haber recibido una autorización previa.

G2 La aplicación debe desarrollarse para ser instalada en el Departamento de Matemáticas y Computación de la Universidad de la Rioja.

- **Gestión de seguridad.** Se comprobará que los usuarios hayan iniciado sesión correctamente y puedan acceder a cada uno de los recursos. También se permitirá acceder a determinadas partes de la aplicación mediante un código.

S1 Permitir el acceso únicamente a los usuarios registrados en la aplicación, permitiendo el acceso excepcionalmente a algunas partes de la aplicación mediante un código, el cual estará codificado.

S2 En caso de que un usuario olvide su contraseña, se le enviará un correo electrónico con un código válido durante 24 horas para regenerar su contraseña.

2.6 Requisitos funcionales

- **Gestión de personas.** Se darán de alta tanto los alumnos como los docentes mediante un formulario asignándoles el rol correspondiente.

P1 La aplicación dispondrá de dos perfiles básicos: alumno y director de proyecto.

P2 Tanto los alumnos como los directores del TFG se darán de alta mediante un formulario.

P3 Una persona solo podrá tener el perfil de alumno o de director de proyecto.

- **Gestionar TFG.** Gestión de los datos del TFG. Esta parte se encargará de guardar todos los datos que forman el perfil del TFG (área, tipo de proyectos, tecnologías implementadas, directores y a qué alumno le corresponde cada TFG).

T1 Todos los TFG deben asignarse a un área y a un tipo de proyecto, los cuales deberán ser introducidos previamente por un administrador.

T2 Cada TFG podrá tener asignada 0 o más tecnologías. En caso de que una tecnología no se encuentre dada de alta, la podrá añadir el alumno.

T3 Un TFG estará asignado a un único alumno.

T4 Un TFG podrá tener 1 o más directores. En caso de que el alumno los introduzca mal o haya un cambio de director, será realizado por el administrador del sistema.

T5 Cada TFG dispondrá de una colección de documentos, los cuales serán aportados por el alumno del TFG o por sus directores.

T6 En caso de que el TFG de un alumno sea desarrollado en una empresa, el alumno será el encargado de introducir los datos de la empresa.

- **Gestión de planificación.** Cada alumno deberá realizar al menos una planificación de su proyecto. En caso de que la planificación sea errónea,

la irá corrigiendo y anotando en su TFG. Para cada replanificación se guardará un histórico con las diferentes planificaciones realizadas.

L1 Todos los TFG deben tener al menos una planificación.

L2 Cada planificación tendrá 4 puntos de control, añadiendo el punto final del proyecto.

L3 El alumno al realizar una replanificación podrá modificar todas las horas de control en los diferentes puntos, aunque la fecha de control sea anterior a la fecha de replanificación.

L4 La fecha de inicio y fin no podrá estar entre los puntos de control.

L5 Las horas en un punto de control no podrán ser mayores que las horas finales planificadas.

L6 Los datos de una replanificación podrán ser modificados el mismo día en el que se introduzcan, pasado el día será una nueva replanificación. No podrá haber 2 replanificaciones en el mismo día.

L7 La planificación solo podrá ser modificada entre la fecha de inicio y de fin del proyecto.

L8 Solo el alumno tendrá acceso a los datos de las replanificaciones. El tutor tendrá acceso a los datos a través de una gráfica de comparación.

- **Gestión de seguimiento.** El alumno irá anotando periódicamente los avances realizados en su TFG. De esta manera, se controlarán las horas invertidas totales hasta la fecha, el porcentaje completado de las diferentes partes que componen el TFG y las horas invertidas en cada uno de los puntos.

D1 Todos los TFG deben tener al menos un punto de seguimiento, siendo recomendable que se introduzca un punto de seguimiento cada 15 días.

D2 Para cada seguimiento, se guardarán las horas totales y horas dedicadas a planificación, análisis, diseño y producto, así como el porcentaje completado de las tareas anteriormente enunciadas. El porcentaje no podrá ser superior a

100 ni inferior a 0 y la suma de las horas parciales debe ser igual a la suma total de las horas.

D3 Solo podrá introducirse el seguimiento entre la fecha de inicio y la fecha de fin del proyecto.

D4 Los datos del progreso del TFG podrán ser modificados el mismo día en caso de que se produzca algún error; en otro caso, se considerará como un nuevo punto de progreso. No podrá haber 2 puntos de progreso en el mismo día.

D5 Solo el alumno tendrá acceso a su propio seguimiento.

- **Gestión de estadísticas.** Esta parte dará una serie de estadísticas del progreso de cada uno de los TFG, así como datos globales.

E1 A las estadísticas tendrán acceso los alumnos y los directores, siempre que hayan seleccionado uno de sus tutorados previamente.

Se crearán los siguientes gráficos:

E2 Se podrán comprobar las diferentes replanificaciones realizadas por un alumno, y también se verán los cambios de hora, en los diferentes puntos de control para cada replanificación.

E3 Se mostrará el proceso de seguimiento, comparado con la última planificación. En el gráfico, se podrá ver en cada punto de progreso introducido las horas reales empleadas y las horas planificadas.

E4 Aparecerá el número de horas totales en cada control de seguimiento desglosadas por tareas.

E5 Se presentará el porcentaje completado de cada tarea en cada punto de seguimiento.

E6 Se podrá ver el reparto de horas invertidas en el último punto de seguimiento, así como un gráfico de barras con el porcentaje completado actualmente de cada tarea, además de mostrar las horas planificadas y reales en dicho punto.

E7 Se mostrarán las horas reales y planificadas del TFG en cuestión junto con la media de los TFG. Además, aparecerá el porcentaje completado de las tareas del proyecto junto con la media de los TFG.

E8 Se expondrá la relación entre horas y porcentaje completado de cada una de las tareas mediante gráficos de puntos.

- **Gestión de alertas.** Cada usuario de la aplicación podrá configurar qué avisos quiere recibir en su correo electrónico y su aceptación para recibir avisos de sus propias acciones.

A1 Cada usuario decidirá si recibe en su correo los avisos de sus propias acciones o no.

A2 Cada usuario podrá personalizar las alertas que recibirá en cada momento; según el tipo de acción desatada, estarán todas activadas por defecto.

- **Gestión de repositorio.** Esta es la parte central de la aplicación. Su cometido será el de gestionar y centralizar todo el proceso de compartición de documentos entre los tutores del TFG y los alumnos, junto a los comentarios que cada uno quiera hacer de las diferentes entregas.

R1 Un nuevo documento deberá ser asignado a un único tipo de documento.

R2 Al realizar una subida al repositorio el documento solo podrá ser asignado a un único TFG.

R3 Los documentos podrán ser descargados por cualquier persona que disponga del código de dicho documento.

R4 Cada documento podrá tener 0 o más comentarios de los directores del TFG o del alumno.

R5 Se establecerá un tamaño máximo de archivo de 150MB. En caso de ser necesario, se podrá modificar dicho tamaño.

3. DISEÑO DEL SISTEMA

Se utilizará una particularización de la arquitectura tradicional de 3 capas: el patrón Modelo Vista Controlador, en adelante MVC. Consiste en 3 tipos de objetos: el modelo que es el objeto de la aplicación, la vista que es su representación y el controlador que define el modo en que la interfaz reacciona a la entrada del usuario. El controlador controla el flujo de la aplicación, pide al modelo aquello que el usuario solicita, y le devuelve (al usuario) una representación del modelo a través de la vista. De esta manera, la lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador.

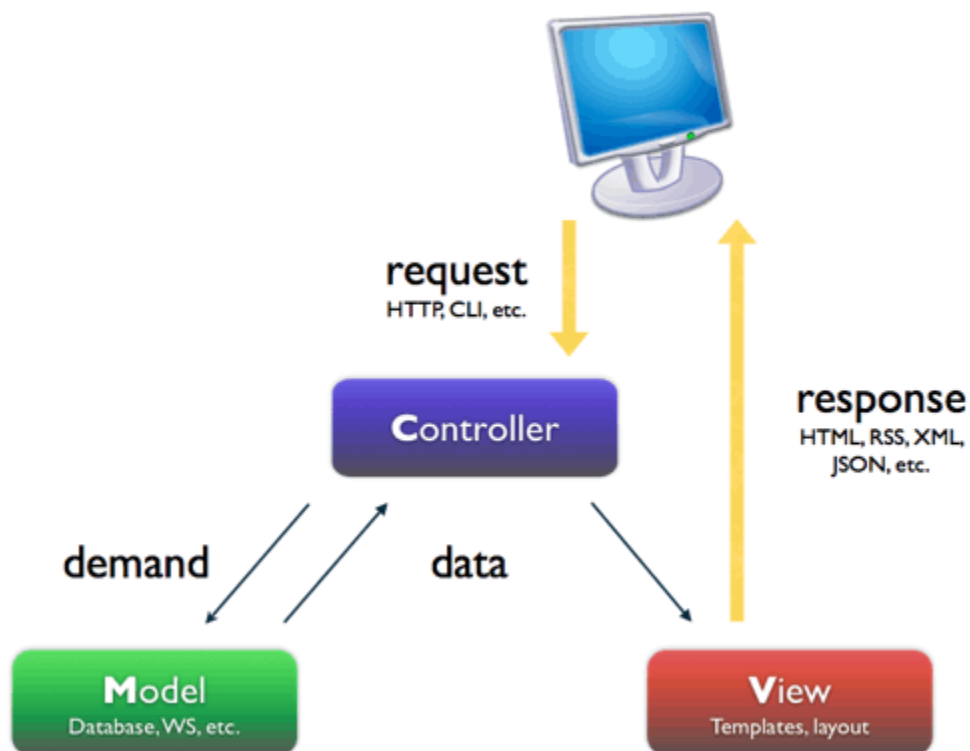


Figura: Modelo MVC

3.1 Diagrama de clases

Una vez que se tienen los casos de uso y los requisitos, es el momento de realizar las clases que formarán la aplicación.

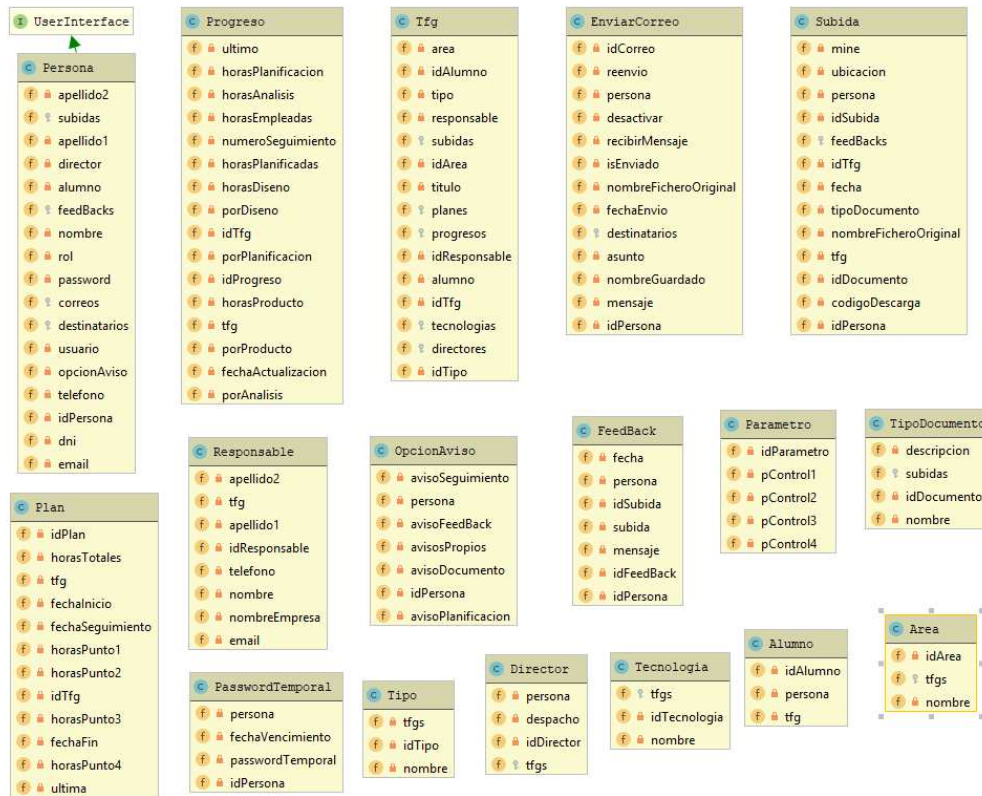


Figura: Diagrama de clases I

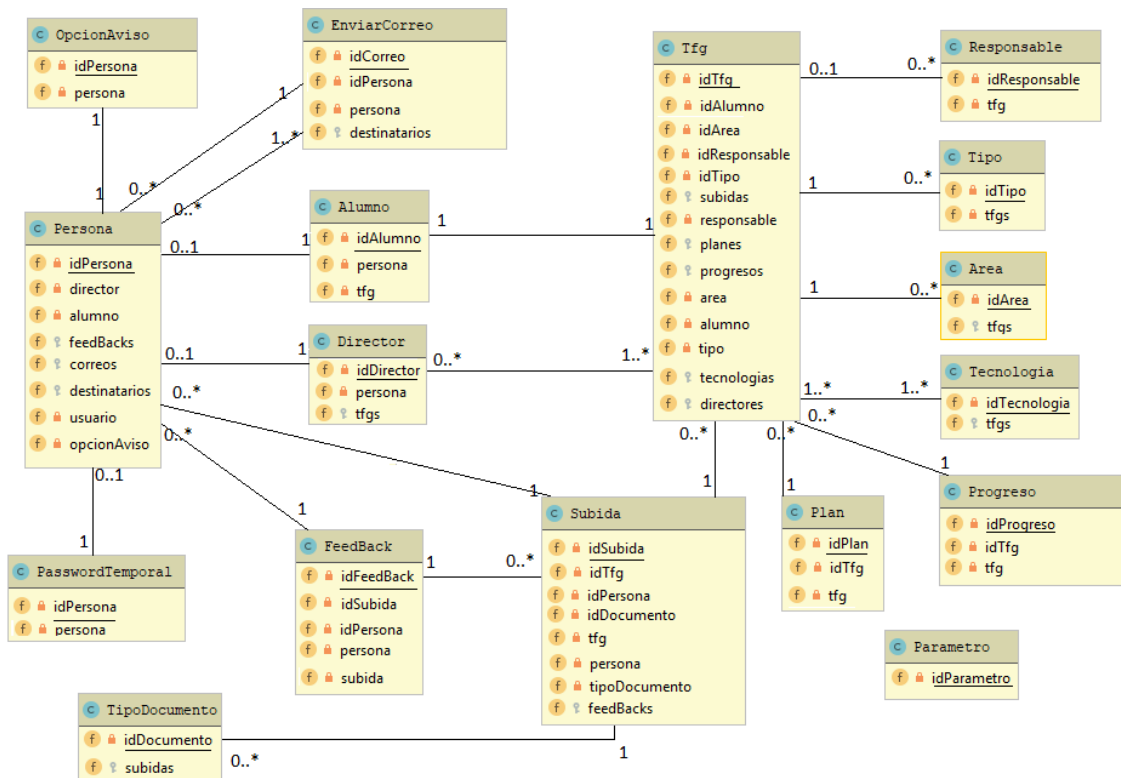


Figura: Diagrama de clases II

Los métodos `getUserName`, `getSalt`, `getRoles` y `eraseCredentials` en persona sirven para gestionar el acceso a la aplicación y forman parte de la interfaz `UserInterface`, que hay que implementar obligatoriamente en Symfony cuando se trabaja con autenticación de usuarios en BD.

3.2 Diseño de datos

A partir del diagrama de clases, se obtiene el siguiente diseño lógico relacional de la base de datos en el cual se muestran las tablas surgidas. Destacar que las tablas han sido generadas por el ORM Doctrine a partir de las entidades que se han creado en Symfony. Conviene mencionar que Doctrine ha creado tablas intermedias en las relaciones N:N. Estas relaciones las ha convertido en 2 relaciones 1:N. Por ejemplo, Doctrine ha creado la tabla `TFGS_directores`, desde la relación entre TFG y director ya que un director puede dirigir en 0 o N Trabajos Fin de Grado y un TFG puede ser dirigido por 1 o más directores.

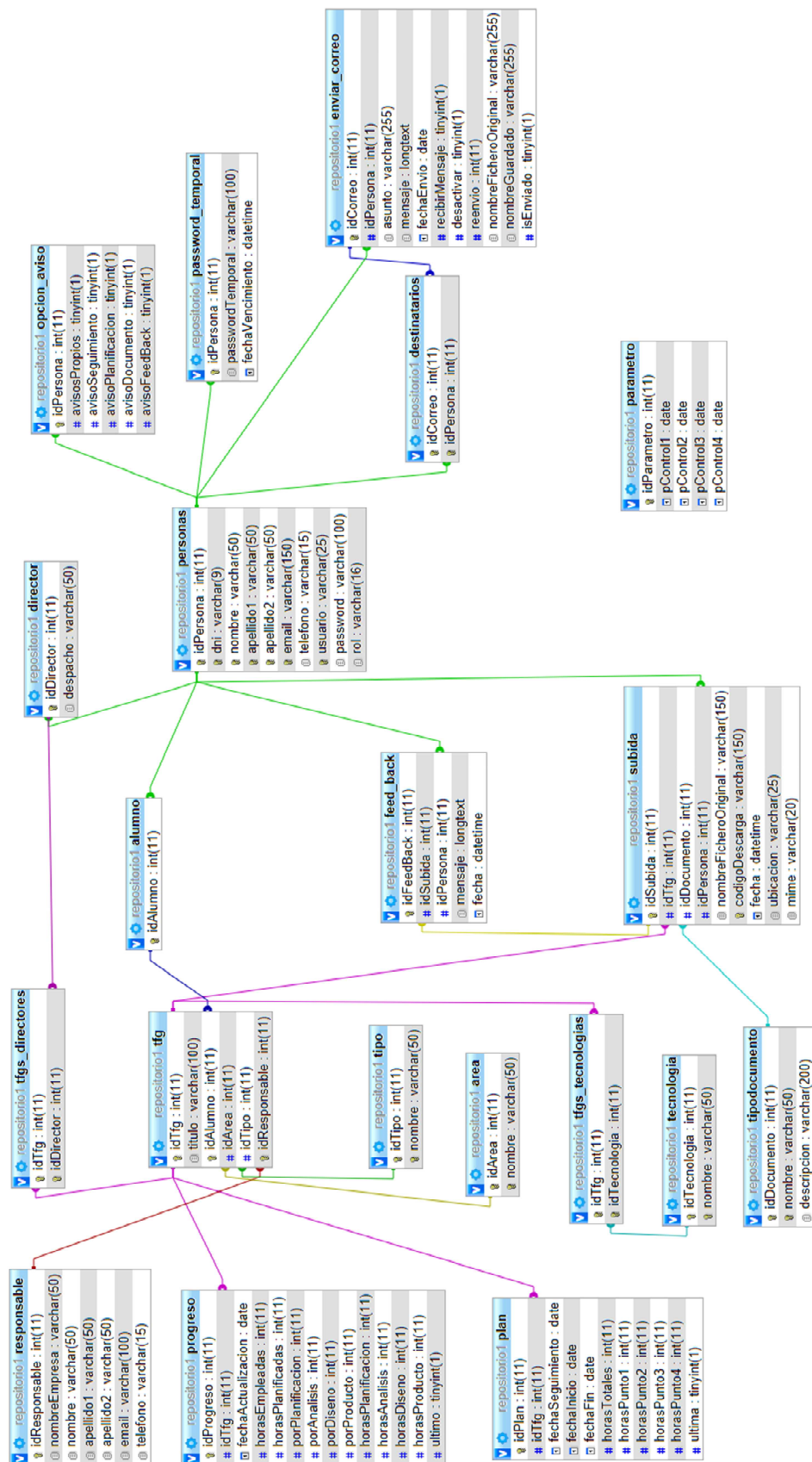


Figura: Base de datos

3.3 Requisitos de datos

De las personas que van a utilizar la aplicación (alumnos y directores) se van a guardar su DNI, nombre, apellido1, apellido2, correo electrónico, teléfono, nombre de usuario, contraseña y su rol. Además, de los directores, se guardará información de su departamento.

Se creará una tabla para gestionar la recuperación de contraseñas por parte del usuario que contendrá el identificador del usuario, fecha de vencimiento y token para regenerar la contraseña.

Además de los datos genéricos, se guardarán los datos de configuración de alertas por un lado, si quieren recibir los avisos que ellos mismo generan; y por otro lado, las diferentes alertas configurables (planificación, seguimiento, documentos y feedback).

Para dar soporte al perfil del TFG, se crearán las clases área, tipo de proyecto y tecnologías, las cuales tendrán un identificador y un nombre, aparte se creará una clase responsable para el caso en el que un TFG se realice en una empresa para lo cual se guardarán: nombre de la empresa, nombre, apellido1 y apellido2 del responsable, teléfono y correo electrónico. Una vez que se tienen todas las tablas auxiliares, ya se pueden definir los datos del TFG que estarán compuestos por: el título, alumno que está haciendo el TFG, directores de ese TFG, área, tipo de proyecto y tecnologías empleadas y el responsable, en caso de tenerlo.

Se contará con una tabla de parámetros donde se guardarán los cuatro puntos de control de la planificación.

Otra parte importante de la aplicación es la referente a la planificación y al seguimiento. Para ello, en la planificación se guardarán los siguientes datos: identificador del TFG, fecha de seguimiento, fecha de inicio y fin del TFG, horas totales estimadas y horas estimadas en los diferentes puntos de control. Para el progreso, se guardará la siguiente información: identificador del TFG; fecha de actualización; horas totales empleadas; porcentaje actual de planificación, análisis, diseño y producto; horas invertidas en planificación, análisis, diseño y

producto y un campo donde estén las horas estimadas en ese punto, calculado a partir de las planificaciones.

Otra zona independiente de la anterior es el repositorio. Para empezar, se contará con una tabla con los diferentes tipos de documentos que serán generados por alumnos y directores, donde aparecerán los campos nombre y descripción.

Cada uno de los documentos que se suban al repositorio, tendrán la siguiente información: proyecto al que hacer referencia, quién ha subido el documento, cuándo lo ha subido, el tipo de documento que es, nombre del documento original, ubicación donde se guarda, tipo mine y código de descarga.

Para concluir, en la parte de datos aparecen los comentarios que se generan en cada documento. Para ello, se debe guardar la siguiente información: código del documento, quién lo ha subido, cuándo lo ha subido y el contenido del comentario.

Y para concluir, en el envío de correos, se deberá guardar la siguiente información: código del correo, persona remitente, destinatarios, asunto, mensaje, nombre del fichero adjunto, nombre con el que se guarda, fecha de envío, recibir copia el remitente en su correo, desactivar el envío de un correo, reenvío cada X días y si ya ha sido enviado.

3.4 Diseño de interfaces

Un punto importante en cualquier aplicación es el aspecto visual que tendrá dicha aplicación. Es importante que resulte atractivo, intuitivo y fácil de utilizar para los usuarios. Otra característica es que se pueda utilizar en diferentes dispositivos electrónicos (ordenadores, teléfonos, tabletas...).

Se ha utilizado Bootstrap que es un framework que permite crear interfaces adaptables al dispositivo utilizado. Para ello hace uso de CSS y JavaScript. Esta técnica de diseño y desarrollo se conoce como *Responsive Design*.

A continuación, se muestran algunas de las pantallas más significativas de la aplicación. A través de dichas pantallas, se puede hacer una idea del diseño y funcionamiento de la aplicación.

3.4.1 Interfaz del menú de la aplicación del director

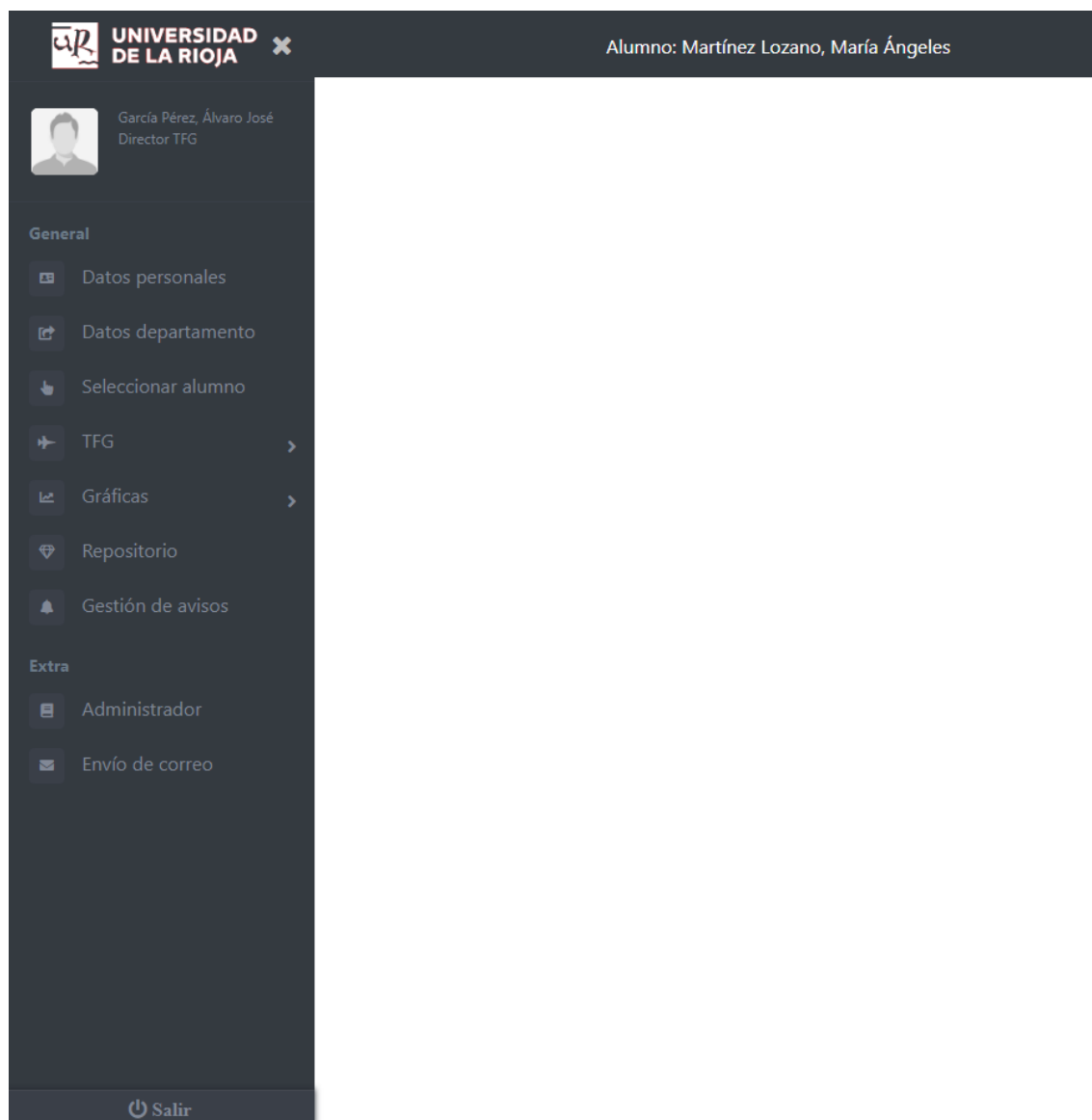


Figura: Interfaz – Menú de la aplicación del director

En esta interfaz, se puede observar el menú de navegación del que dispondrán los directores.

3.4.2 Interfaz del menú de la aplicación del alumno

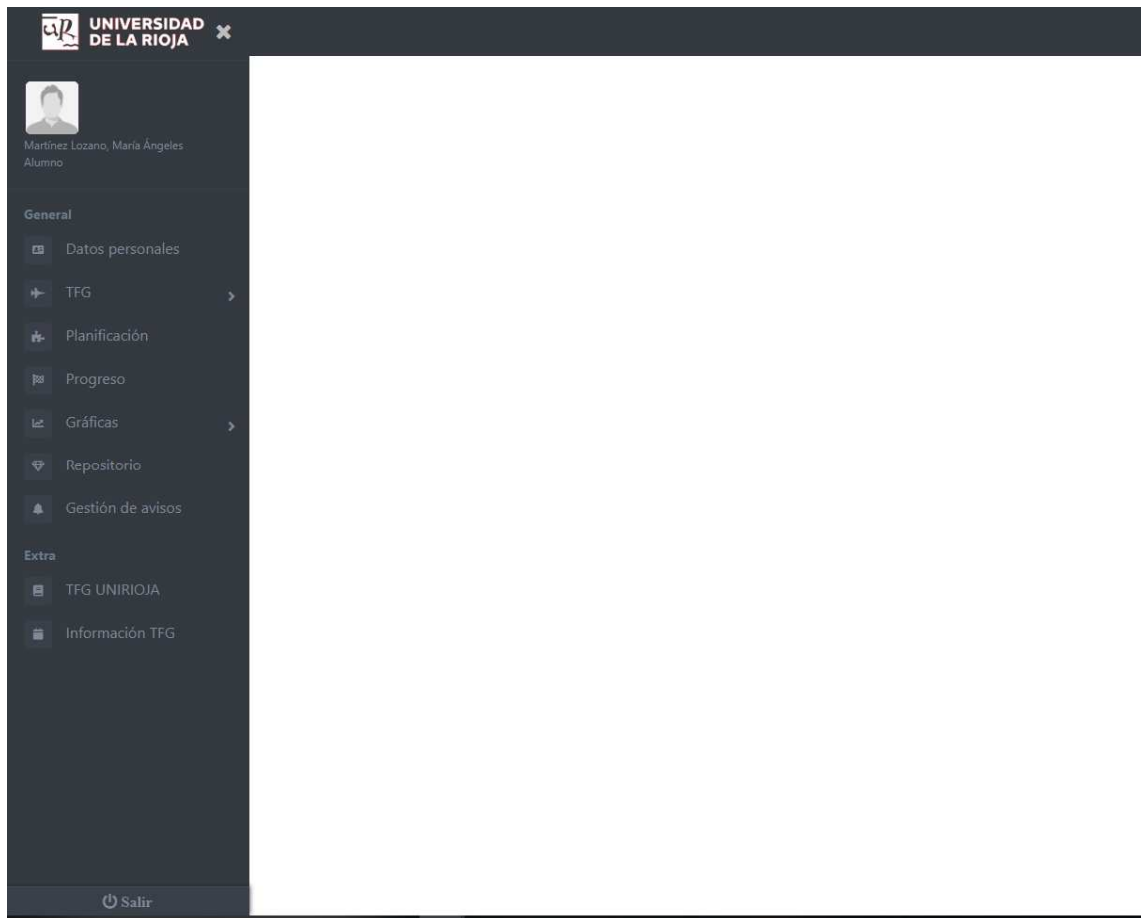


Figura: Interfaz – Menú de la aplicación del alumno

En esta interfaz, se puede observar el menú de navegación del que dispondrán los directores.

3.4.3 Acceso a la aplicación

La primera pantalla que aparece siempre al acceder es la de login. En este caso, se nos pedirá un nombre de usuario y una contraseña. También es recomendable dar la posibilidad al usuario de recuperar la contraseña en caso de olvido, y que los nuevos usuarios puedan darse de alta.

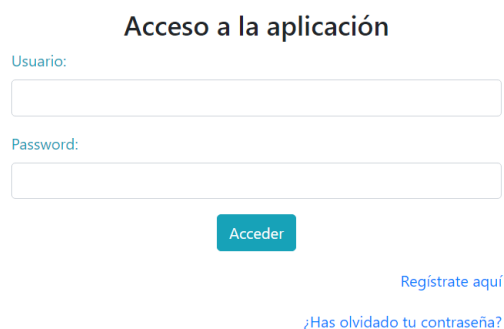


Figura: Interfaz – Validación de usuarios

3.4.4 Recuperación de contraseñas

En el caso de que un usuario olvide su contraseña o quiera cambiarla, se le dará la posibilidad de hacerlo a través del enlace ¿Has olvidado tu contraseña? Posteriormente se le pedirá el DNI y el correo electrónico. En caso de que estén dichos datos, se le enviará un correo electrónico personalizado con las indicaciones. En caso contrario, se le mostrará un error de que no existen dichos datos en el sistema.



Figura: Interfaz – Recuperación de contraseña I

Hola García Pérez, Álvaro José

Solicitaste un restablecimiento de la contraseña para tu cuenta de usuario (alvaro).

Por favor, haz clic en el botón de abajo para cambiar tu contraseña.

[Cambiar tu contraseña](#)

Tenga en cuenta que este enlace es válido durante 24 horas. Una vez que el límite de tiempo haya caducado, deberá volver a enviar la solicitud para restablecer la contraseña.

Figura: Interfaz – Recuperación de contraseña II

En la figura anterior, se puede ver la plantilla que se les enviará a los usuarios para regenerar la contraseña, junto a un link con el token que hará posible el cambio de contraseña.

Establecer contraseña

Hola Álvaro José García Pérez, indica tu nueva contraseña.

La contraseña debe incluir de 6 a 10 caracteres, incluyendo mayúsculas, minúsculas, números y caracteres especiales.

Nueva contraseña:

Nueva contraseña (de nuevo):

[Cambiar contraseña](#)

[Cancelar](#)

Figura: Interfaz – Recuperación de contraseña III

El siguiente paso con el que se encontrarán los usuarios será el de introducir la nueva contraseña. Para ello, se deberá introducir en los dos campos una contraseña de 6 a 10 caracteres en los cuales debe haber al menos una mayúscula, una minúscula y un dígito.

3.4.5 Alta de alumnos o directores

Alta Alumno

DNI	<input type="text" value="DNI"/>
Nombre	<input type="text" value="Nombre"/>
Apellido 1	<input type="text" value="Apellido 1"/>
Apellido 2	<input type="text" value="Apellido 2"/>
Correo electrónico	<input type="text" value="Correo electrónico"/>
Teléfono	<input type="text" value="Teléfono"/>
Identificador de usuario	<input type="text" value="Usuario"/>
Contraseña	<input type="password" value="Contraseña"/>
Repite contraseña	<input type="password" value="Contraseña"/>

Figura: Interfaz – Alta de Alumno

Otra parte importante es el alta de alumno y director. Para ello, se mostrará un formulario con todos los datos necesarios. Para el director, se añadirá el departamento.

3.4.6 Alta/Modificación de datos del TFG

Perfil TFG

Título	<input type="text" value="Gestión y planificación de la evaluación de los alumnos de un centro escolar"/>
Área	<input type="text" value="Gestión"/>
Tipo	<input type="text" value="Tipo 111"/>
Tecnologías	<input type="text" value="Android"/> <input type="text" value="DB2"/> <input type="text" value="Doctrine"/> <input type="text" value="Java"/>
Otra tecnología	<input type="text" value="Otra tecnología que no esté en la lista"/>
Directores del TFG	<input type="text" value="Abad Pérez , Juan"/> <input type="text" value="García Pérez , Álvaro José"/> <input type="text" value="García Pérez , David"/>

Figura: Interfaz – Modificación datos del TFG

En este formulario, el alumno completará todos los datos relativos a su TFG, podrá modificarlo las veces que considere oportunas; pero una vez que rellene los datos relativos a su director de TFG, no podrá modificarlo.

3.4.7 Añadir/Modificar representante de la empresa

Modificar datos de la persona responsable de la empresa

Nombre de la empresa	<input type="text" value="Emesa S.A"/>
Nombre	<input type="text" value="Eduardo"/>
Apellido 1	<input type="text" value="Pérez"/>
Apellido 2	<input type="text" value="Martínez"/>
Teléfono	<input type="text" value="941252525"/>
Correo electrónico	<input type="text" value="eduardo@emesa.com"/>

Figura: Interfaz – Modificar datos responsables de la empresa

Esta interfaz solo será usada en el caso de que a un alumno le dirijan su TFG desde una empresa. Para ello, el alumno completará los datos del responsable que le tutoriza el TFG en la empresa.

3.4.8 Búsqueda de TFG

Búsqueda de TFG por perfiles

Área	<input type="text"/>
Tipo de proyecto	<input type="text"/>
Tecnología	<input type="text"/>

Figura: Interfaz – Búsqueda de proyectos por perfil

Mediante esta interfaz, se podrán buscar los TFG que cumplan unos criterios determinados por el usuario.

3.4.9 Añadir/Modificar replanificación

Añadir plan de seguimiento

Fecha de inicio 1 mar. 2019

Fecha de fin 28 jun. 2019

Horas totales 300

Horas invertidas en cada punto de control

15/03/2019	30
15/04/2019	60
15/05/2019	180
15/06/2019	220

Aceptar
Cancelar

Figura: Interfaz – Añadir plan de seguimiento

Mediante esta interfaz, el alumno podrá introducir las replanificaciones que tenga su proyecto. En caso de que se equivoque al introducir los datos, podrá modificarlos antes de las 23:59 de ese mismo día.

3.4.10 Añadir/Modificar progreso actual del TFG

Añadir progreso de seguimiento

Horas invertidas en cada apartado

Nº de horas de planificación	50
Nº de horas de análisis	50
Nº de horas de diseño	50
Nº de horas de producto	158

Porcentaje concluido en cada apartado

% Concluido de planificación	100
% Concluido de análisis	100
% Concluido de diseño	100
% Concluido de producto	100

Aceptar
Cancelar

Figura: Interfaz – Añadir punto de seguimiento

En esta interfaz, lo que se busca es introducir en el sistema el estado actual de los diferentes TFG. Para ello, el alumno introducirá los datos. Se recomienda que cada 15 días introduzca un seguimiento.

3.4.11 Repositorio

La parte del repositorio es una de las más importantes. A través de ella, se centrará la mayor parte de la comunicación entre el director del TFG y el alumno.

Ha sido diseñada dándole un aspecto de foro.

REPOSITORIO PRINCIPAL <<Descarga ZIP>>				
TIPO DE DOCUMENTO	Nº Archivos	Nº Mensajes	NOMBRE DE ARCHIVOS	ÚLTIMO MENSAJE
Memoria La memoria es un documento en el que quedan recogidos los resultados y el desarrollo del proyecto.	1	1	Nombre del documento: MemoriaMayo.docx Subido por: Martínez Lozano, María Ángeles TFG de: Martínez Lozano, María Ángeles Hora: 09:50:06 21/07/2019 <<DESCARGAR>>	Mensaje: Presta especial atención a la ortografía Autor: García Pérez, Álvaro José Hora del mensaje: 09:51:16 21/07/2019 <<COMENTARIOS>>
Acta Documento escrito en el que se relaciona lo sucedido, tratado o acordado en una junta o reunión.	0	0		
DOP Documento de objetivos del proyecto.	0	0		
Otros Otros tipo de documentos.	0	0		

Figura: Interfaz – Repositorio I

La primera pantalla con la que se encontrará el usuario será aquella donde se vean los diferentes tipos de documento junto con una estadística del número de archivos y el número de mensajes que se ha generado, así como el último documento en el que se hayan producido cambios, ya sea porque se ha subido recientemente o se ha añadido un nuevo comentario. Desde esta interfaz, se podrá descargar un documento o acceder a los documentos que se hayan generado en un tipo de documento.

REPOSITORIO (Memoria)		
DOCUMENTO	MENSAJES	ÚLTIMO MENSAJE
Nombre del documento: MemoriaMayo.docx Subido por: Martínez Lozano, María Ángeles TFG de: Martínez Lozano, María Ángeles Hora: 09:50:06 21/07/2019 <<DESCARGAR>>	1	Mensaje: Presta especial atención a la ortografía Autor: García Pérez, Álvaro José Hora del mensaje: 09:51:16 21/07/2019 <<COMENTARIOS>>

Archivo: Ningún archivo seleccionado

Figura: Interfaz – Repositorio II

En esta interfaz, aparecerán los documentos que se tienen de un tipo de documento así como el último comentario generado. Desde esta interfaz, se podrán subir nuevos documentos. En el caso de que sea el tutor y no tenga un alumno seleccionado, deberá indicar a qué alumno le sube el documento.

REPOSITORIO - (Memoria) - MemoriaMayo.docx <<DESCARGAR>>	
TFG: Gestión y planificación de la evaluación de los alumnos de un centro escolar TFG de: Martínez Lozano, María Ángeles	
MENSAJES	USUARIO
Presta especial atención a la ortografía	Autor: García Pérez, Álvaro José Hora del mensaje: 09:51:16 21/07/2019

Figura: Interfaz – Repositorio III

Por último, solo queda la interfaz a través de la cual se podrán ver todos los comentarios que se han generado sobre un documento concreto.

3.4.12 Modificar opciones de aviso al correo electrónico

Modificar opciones de aviso

- Avisos propios ☒
- Avisos de seguimiento ☒
- Avisos de planificación ☒
- Avisos de nuevos documentos ☒
- Avisos de nuevos comentarios ☒

Figura: Interfaz – Opciones de aviso

Hay que destacar la importancia de que tanto los alumnos como los directores de TFG reciban avisos cuando se produzca una alteración en el proyecto. Por eso, se les da la posibilidad de que autoconfiguren los mensajes de aviso que quieren recibir.

3.4.13 Selección de alumnos

Seleccionar alumno		
	Alumno	Título del TFG
<input type="radio"/>	García Martínez, Daniel	Localización de objetos en imágenes mediante técnicas de aprendizaje profundo
<input type="radio"/>	Martínez Lozano, María Ángeles	Gestión y planificación de la evaluación de los alumnos de un centro escolar
<input type="radio"/>	No seleccionar ninguno	

Figura: Interfaz – Selección de alumnos

Mediante la anterior interfaz, el director del TFG seleccionará a uno de sus alumnos para poder comprobar los avances de ese alumno en particular.

3.4.14 Envío de notificaciones

Dar de alta nuevo mensaje de correo electrónico

Enviar a:
Martínez Lozano, María Ángeles

Asunto:

Mensaje:

Adjuntar fichero: Ningún archivo seleccionado

Fecha de envío del mensaje:

¿Reenviar periódicamente el mensaje? (Días):

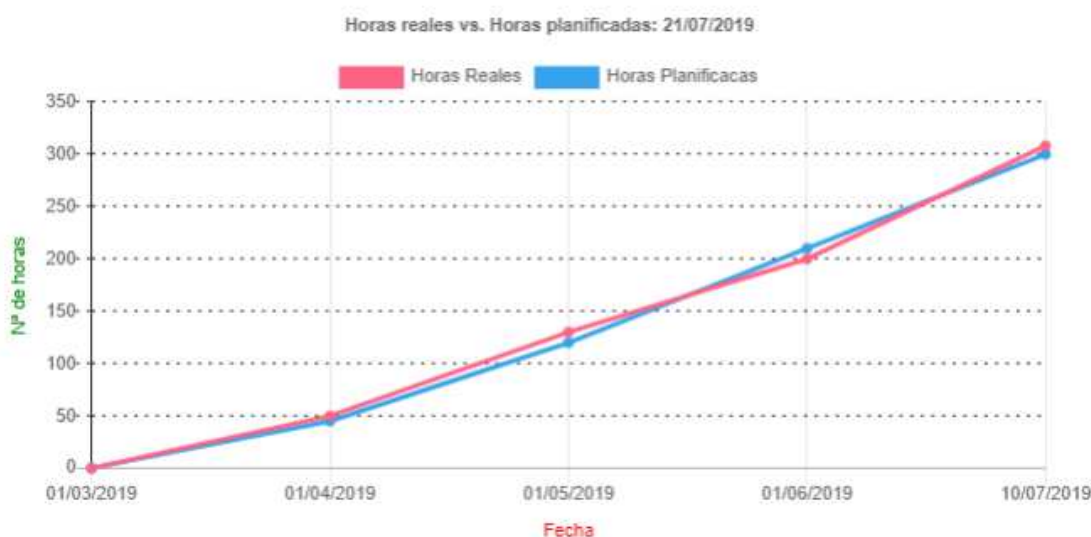
Enviarme copia: ☐

Figura: Interfaz – Envío de notificaciones

Mediante esta interfaz, el director del TFG podrá enviar mensajes a sus alumnos. Podrá enviárselos en el momento si la fecha de envío es la fecha del día o en un futuro y de esa manera tener mensajes programados para ser lanzados en un día concreto. También dispondrá de la opción para enviar mensajes cada X días.

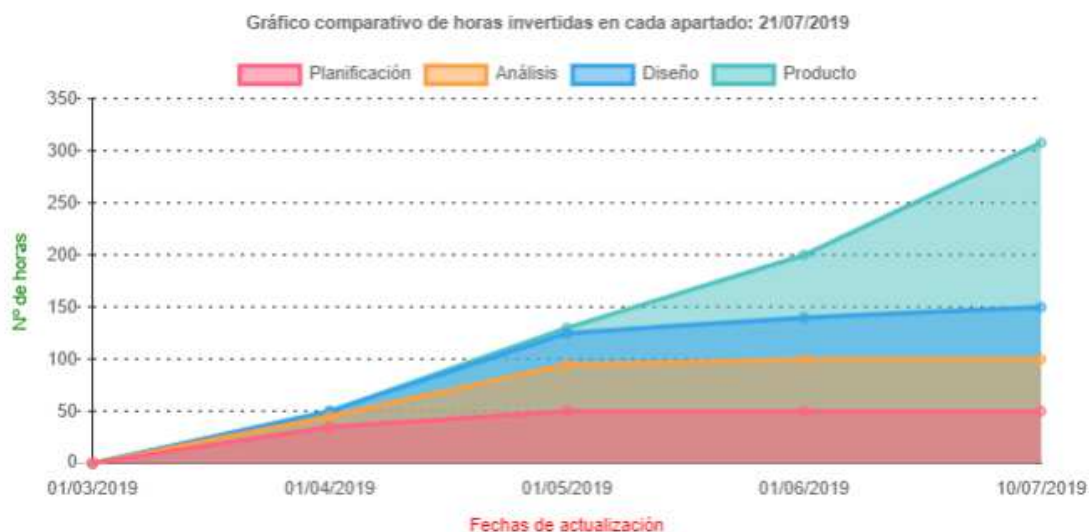
3.4.15 Gráficos

A continuación, se muestran algunos de los gráficos que aparecen en la aplicación. Como se podrá ver a continuación, se presenta un gráfico de cada uno de los tipos generados (línea, barras o puntos).



En este gráfico se comparan las horas planificadas con las horas reales invertidas en el TFG. Para ello, se usan los datos de la última planificación y los distintos puntos de seguimiento que se han introducido a lo largo del TFG. Por ello, es importante que se actualice regularmente el estado del TFG.

Figura: Interfaz – Gráfico comparativo del progreso vs. planificación



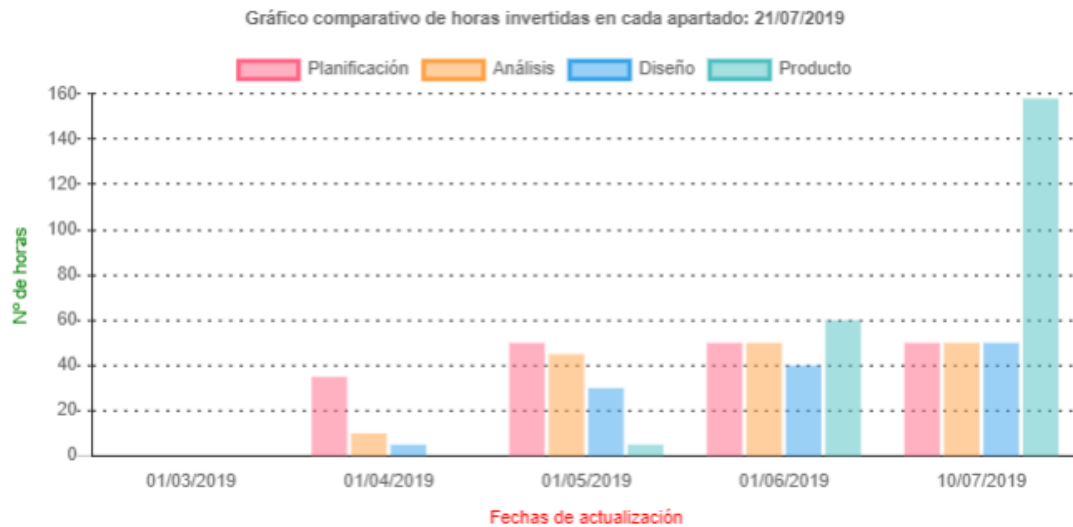
En el gráfico anterior, se puede ver la evolución de las horas invertidas en cada punto de seguimiento. Como se puede observar en el gráfico, se acumulan las horas de cada parte del TFG.



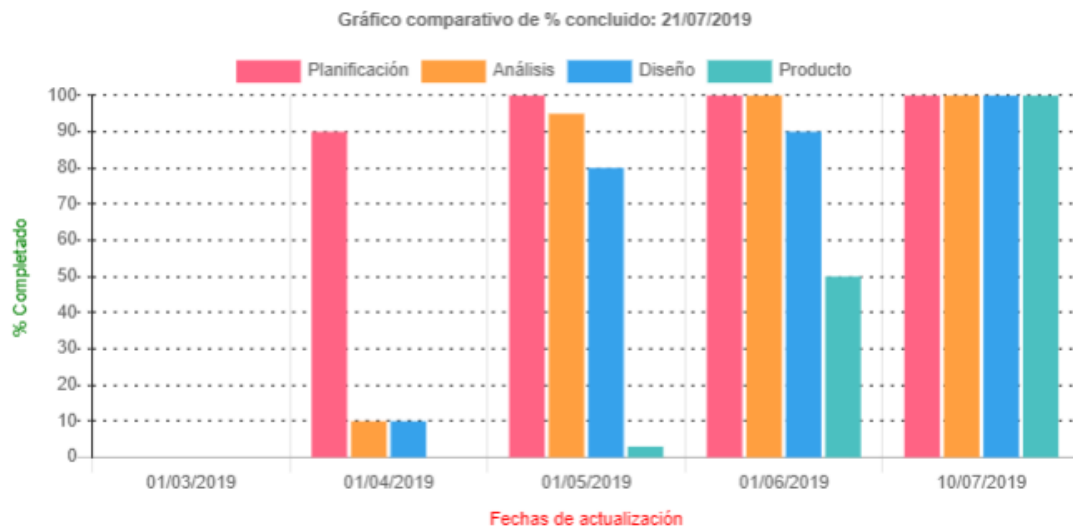
Mediante este gráfico, se puede ver el porcentaje concluido en cada parte del proyecto. Los valores recogidos estarán entre 0 % y 100 %. Se llegará al 100 % cuando se concluya dicha parte del TFG.

Figura: Interfaz – Gráficos de evolución del progreso (Líneas)

Mediante estos gráficos, se le dará al alumno y al director del TFG una visión de cuál es el estado del proyecto en el que se está trabajando y se dará la posibilidad de verlo en formato línea o barras.



En el gráfico anterior, se puede ver la evolución de las horas invertidas en cada una de las partes del TFG.



Mediante este gráfico, se puede ver el porcentaje concluido en cada parte del proyecto. Los valores recogidos estarán entre 0 % y 100 %. Se llegará al 100 % cuando se concluya dicha parte del TFG.

Figura: Interfaz – Gráficos de evolución del progreso (Barras)



En el gráfico anterior, se puede observar la relación entre las horas totales invertidas y el % concluido de la planificación, tanto del TFG como de la media de los demás compañeros que están realizando su TFG.



En el gráfico anterior, se puede observar la relación entre las horas totales invertidas y el % concluido del análisis, tanto del TFG como de la media de los demás compañeros que están realizando su TFG.

Figura: Interfaz – Gráficos de puntos

En este gráfico, se puede observar la comparación entre el porcentaje concluido y las horas invertidas tanto de un proyecto como la media de los demás proyectos.

4. IMPLEMENTACIÓN

4.1 Introducción

Se dedicará el capítulo de implementación a explicar cuáles han sido las tecnologías utilizadas para la realización del proyecto, así como a relatar alguna de las dificultades más importantes encontradas y sus soluciones. Como se irá viendo en este documento, se han buscado soluciones a los problemas surgidos.

Se ha implementado toda la funcionalidad que se enumeró en el punto 2.3 Funcionalidad de la futura aplicación, a excepción de la publicación en la red social Facebook; debido a un cambio en sus condiciones ya no es posible publicar en un grupo mediante un correo electrónico. Desde octubre de 2018, se exige un token para publicar en páginas y hay que pedir previa autorización a Facebook, además de entregar una serie de documentación. Por todo ello, no se ha implementado dicha funcionalidad.

Hubo un problema al comienzo del TFM por malas elecciones relacionadas con el framework y el ORM, las cuales provocaron la pérdida de un gran número de horas y que casi lleva a abortar este TFM.

4.2 Tecnologías empleadas

La arquitectura software de la aplicación se basa en el empleo de tres capas diferenciadas: presentación, negocio y persistencia. Seguidamente, se enumeran las tecnologías empleadas en cada capa:

Capa de presentación

- Framework Symfony
- Twig
- Bootstrap
- CSS
- Java Script
- Ajax

- Librería charts.js
- HTML

Capa de negocio

- Symfony
- Swiftmailer

Capa de persistencia

- Doctrine
- MariaDB Gestor de Base de datos

A continuación, se destacan algunas de las situaciones a modo de ejemplo donde el uso de las tecnologías ha ayudado al desarrollo de la aplicación, así como la solución a alguno de los problemas surgidos durante la aplicación.

4.2.1 Doctrine

Este ORM nos aísla de la capa de persistencia, facilitándonos la tarea. En primer lugar, se deben realizar las anotaciones en las entidades.

```
/**
 * Persona
 */
@ORM\Table(name="personas", uniqueConstraints={@ORM\UniqueConstraint(name="persona_unique", columns={"dni", "nombre", "apellido1", "apellido2"})},
 * indexes={@ORM\Index(name="idx_persona",
columns={"apellido1", "apellido2", "nombre"})})
 * @ORM\Entity(repositoryClass="AppBundle\Repository\PersonaRepository")
 */
class Persona implements UserInterface
{
    /**
     * @var int
     */
    * @ORM\Column(name="idPersona", type="integer")
    * @ORM\Id
    * @ORM\GeneratedValue(strategy="AUTO")
    */
    private $idPersona;
    ..
}
```

Como se ve en el ejemplo, se tiene que indicar mediante la anotación en qué tabla se guardarán los datos de la entidad, así como los índices. Encima de cada atributo, se definirán el nombre del campo, tipo, si es clave primaria...

En el Anexo IV, se puede ver cómo realizar las relaciones entre entidades.

Otro punto importante de Doctrine es cómo maneja la interconexión de datos entre la base de datos y las clases. A continuación, se puede ver un ejemplo de una consulta:

```
$query = $this->createQueryBuilder('t')
    ->select('t','s','max(s.fecha) as fs ,max(f.fecha) as ff,
count(DISTINCT(s.idSubida)) as ns,count(f.idFeedBack) as nf')
    ->leftJoin('t.subidas','s')
    ->leftJoin('s.tfg','p')
    ->where('p.idAlumno = :idAlumno')
    ->setParameter('idAlumno', $idAlumno)
    ->groupBy('t')
    ->orderBy('t.nombre')
    ->getQuery();
return $query->getResult();
```

En este punto hubo un problema: no comprender el funcionamiento del ORM. La consulta anterior sirve para buscar los tipos de documentos, así como las subidas de un proyecto que se localiza por el idAlumno. El caso es que, si en el select no se añade la entidad subida, localizará qué tipos de documentos se han subido, pero cuando se intente acceder a la colección de subidas, en vez de mostrar las subidas de un proyecto, se cargarán todas las subidas; sin embargo, al poner la entidad en el select ya aparecerán cargadas las subidas de un proyecto concreto.

Otro aspecto importante es el de generar los formularios. Con el framework se facilita la tarea, pero hay ocasiones en las que se requiere reutilizar un formulario y habrá que mostrar diferentes campos. Para ello, se pueden pasar datos en el array, pero tienen que tener un formato específico, así como crear variables en la clase.

```
private $idDirector;
public function buildForm(FormBuilderInterface $builder, array $options)
{
    $this->idDirector=($options['attr']['id']);
    $builder
```

```
->add('destinatarios', EntityType::class,
    [
        'label' => 'Enviar a',
        'multiple' => true,
        'class' => 'AppBundle:Persona',
        'query_builder' => function (PersonaRepository $er) {
            return $er->createQueryBuilder('p')
                ->leftJoin('p.alumno', 'a')
                ->leftJoin('a.tfg', 't')
                ->leftJoin('t.directores', 'd')
                ->where('d.idDirector= :idDirector' )
                ->setParameter('idDirector', $this->idDirector )
                ->orderBy('p.apellido1', 'ASC')
                ->addOrderBy('p.apellido2', 'ASC')
                ->addOrderBy('p.nombre', 'ASC');
        },
        'label_attr' => array('class' => 'col-sm-3 col-form-label
text-right'),
        'attr' => array('class' => 'form-control')
    ]
);
..
}
```

En el código anterior se puede ver: cómo se pasa un identificador de persona a través de un array y cómo posteriormente se utiliza para crear fácilmente un campo select con datos personalizados cargados.

4.2.2 Charts.js

Otro aspecto importante en este TFM es la generación de gráficos. Actualmente se utiliza una biblioteca de Google para generar los diferentes gráficos.

A continuación, se puede ver una comparativa de las distintas alternativas con las que se puede contar para realizar gráficos.

	Tipos de gráficas soportadas					Interactividad		Tecnologías	
	Line	Area	Donut	Radar	Gantt	onClick	mouseOver	HTML5	VML
Chart.js	✓	✓	✓	✓	✗	✗	✓	✓	✗
xCharts	✓	✗	✗	✗	✗	✓	✓	✗	✗
canvasXpress	✓	✓	✗	✗	✗	✓	✓	✓	✓
Flot Charts	✓	✓	✓	✗	✗	✓	✓	✓	✗
FusionCharts	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figura: Comparativa de librerías JS para generar gráficas

Una vez estudiadas las diferentes posibilidades, se decide emplear la librería chart.js. Entre sus ventajas se pueden destacar:

- Es adaptativo (*responsive*).
- Modular (Funciones separadas).
- Open Source.
- La documentación de chart.js está bien organizada y provee información detallada de cada una de las características.

4.2.3 Twig

El motor de plantillas ha sido todo un acierto y ha ayudado a la hora de diseñar las diferentes interfaces que componían la aplicación, así como las plantillas para mandar por correo electrónico a los usuarios.

```
<div class="container">
    {{ form_start(form, {'attr': {'id':form.vars.id,'onsubmit':'return
validarFormulario(this);','action':''}} ) }}
    {{ form_errors(form) }}
<div id="altaPlanificacion">
    <h3 class="text-center text-black pt-5">{{ accion }}</h3>

    <div class="form-group row">
        {{ form_label(form.fechaInicio) }}
        <div class="col-sm-4">
            {{ form_widget(form.fechaInicio.day) }}
            {{ form_widget(form.fechaInicio.month) }}
            {{ form_widget(form.fechaInicio.year) }}
            {{ form_errors(form.fechaInicio) }}
        </div>
    </div>
    ..
    <div class="form-group" style="text-align:center;">
        <div class="form-group" style="text-align:center;">

            {% if(modificacion==false) %}
                {{ form_widget(form.altaseguimiento, {'label': 'Aceptar'
                    , 'label_attr': {'class': 'MYCLASSFOR_LABEL'
                    , 'attr': {'class': 'btn btn-info btn-md
MYCLASSFOR_INPUTS'}}) }}
            {% else %}
                {{ form_widget(form.altaseguimiento, {'label': 'Aceptar'
                    , 'label_attr': {'class': 'MYCLASSFOR_LABEL'
                    , 'attr': {'class': ' btn btn-info btn-md
MYCLASSFOR_INPUTS'}}) }}
            {% endif %}

            <a href="{{ path('homepage') }}" class="btn btn-info btn-
md">Cancelar</a>
        </div>
    </div>
```

```
</div>
    {{ form_end(form) }}
</div>
```

Utilizando las plantillas de twig junto con Bootstrap, se consigue generar en pantalla un formulario, se utiliza la etiqueta `{{ form_widget(form.campo) }}` para crear un campo del formulario a través de los parámetros que se han definido en el formulario.

4.2.4 JQuery

```
function validarFormulario() {
    $("#errores").html('');

    isValidado=true;

    {% if (parametro is not defined or parametro!='C') %}
        valorContrasena1=$("#{{form.password.first.vars.id}}").val();
        valorContrasena2=$("#{{form.password.second.vars.id}}").val();
        isValidado=(validaCorreo(valorContrasena1,valorContrasena2));
    {% endif %}

    if(!validateTelefono($("#{{form.telefono.vars.id}}").val())){
        mostrarAviso('Número de teléfono no válido');
        isValidado=false;
    }

    if(!validateEmail($("#{{form.email.vars.id}}").val())){
        mostrarAviso('El correo electrónico no es válido');
        isValidado=false;
    }

    if(!validar_dni_nif_nie($("#{{form.dni.vars.id}}").val())){
        mostrarAviso('El formato del DNI no es correcto');
        isValidado=false;
    }

    return isValidado;
}
```

Se utiliza JQuery para realizar la validación de los formularios en el lado del cliente. Los campos son identificados por Symfony; por lo tanto, se utilizan las etiquetas correspondientes para poder localizarlos.

4.2.5 Swiftmailer

Para la gestión de correos electrónicos, se decidió configurar un servicio, el cual tendrá que ser establecido en el fichero `services.yml`.


```
app.mailer:
class: AppBundle\Services\Mailer
arguments: ["null"]
```

Y se crea el servicio en la carpeta Services con el nombre Mailer.php.

```
class Mailer{
    public function sendEmailAction(\Swift_Mailer $mailer, String $subject, String
    $remiteante, String $apodo, String $contenido, String $correoEmisor, String
    $apodoEmisor) {
        $message = \Swift_Message::newInstance()
            ->setSubject($subject)
            ->setFrom(array($correoEmisor=> $apodoEmisor))
            ->setTo(array($remiteante => $apodo))
            ->setBody($contenido, 'text/html');
        $mailer->send($message);
    }
    public function sendEmailMessageAction(\Swift_Mailer $mailer, $message) {
        $mailer->send($message);
    }
    public function generarCadenaAction(String $largo){
        $cadena_base =
        'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
        //$cadena_cadena_base='!#%^&*()_./<>?:[]{}|\|=+';
        $password = '';
        for ($i=0; $i < $largo; $i++){
            $password .= $cadena_base[rand(0, strlen($cadena_base) - 1)];
        }
        return $password;
    }
}
```

4.2.6 Sonata

Este vendor de Symfony facilita la creación de gestión CRUD (*Create, Read, Update, Delete*) de tablas sencillas, pudiendo personalizarlo con controladores y otras opciones, pero en este caso el tiempo de desarrollo crecerá considerablemente.

```
class AreaAdmin extends AbstractAdmin{
    protected function configureDatagridFilters(DatagridMapper
    $datagridMapper) {
        $datagridMapper
            ->add('nombre');
    }
    protected function configureListFields(ListMapper $listMapper){
        $listMapper
            ->add('nombre')
            ->add('_action', null, [
                'actions' => [
                    'show' => [],
                    'edit' => [],
                    'delete' => [],
                ],
            ],
        ];
    }
    protected function configureFormFields(FormMapper $formMapper){
        $formMapper
            ->add('nombre')
```

```

    ;
}
protected function configureShowFields(ShowMapper $showMapper) {
    $showMapper
        ->add('nombre')
    ;
}
}

```

Todo el código es generado con la instrucción:

```
php bin/console sonata:admin:generate AppBundle/Entity/Area
```

4.3 Seguridad

Se ha tenido especial cuidado en el control de acceso y en la gestión de contraseñas, utilizando para ello las herramientas que ofrece Symfony para su gestión.

4.3.1 Control de acceso

En todas las páginas se comprobará que el usuario haya sido autenticado correctamente y que posea los permisos necesarios, a excepción de aquellas páginas que no necesitaban un usuario. Para ello, se hará uso del componente de seguridad de Symfony.

Este proceso se hará a través del objeto de seguridad en la sesión.

```

if($this->getUser()==null){
    $this->get('session')->getFlashBag()->add('danger', 'Usuario no
logueado');
    return $this->redirect($this->generateUrl('homepage'));
}

```

Además de la comprobación del usuario en la sesión, se comprueba que el usuario tenga permisos para acceder a esa ruta, a través de los roles que ofrece Symfony. Esta configuración se encuentra en el archivo security.yml, lo que simplifica el control de seguridad de la aplicación.

access_control:

```

- { path: /graficas, roles: [ROLE_ALUMNO,ROLE_DIRECTOR] }
- { path: ^/admin, roles: [ROLE_DIRECTOR] }
- { path: ^/enviarCorreoDirector, roles: [ROLE_DIRECTOR] }

```

4.3.1 Gestión de contraseñas

Las contraseñas se guardarán codificadas en la base de datos bajo el algoritmo bcrypt que se codificará 4 veces sobre sí mismo.

```
encoders:  
  AppBundle\Entity\Persona:  
    algorithm: bcrypt  
    cost: 4
```

El proceso para guardar una nueva contraseña cifrada es el siguiente:

```
$factory = $this->get("security.encoder_factory");  
$encoder = $factory->getEncoder($persona);  
$persona->setPassword($encoder->encodePassword($password, $persona-  
>getSalt()));
```

El proceso de login_check lo ejecuta automáticamente Symfony. Para ello, se ha realizado la configuración previa en el archivo security.yml.

```
providers:  
  our_db_provider:  
    entity:  
      class: AppBundle:Persona  
      property: usuario  
main:  
  anonymous: ~  
  
  provider: our_db_provider  
  form_login:  
    login_path: /login  
    check_path: /login_check  
  logout:  
    path: /logout  
    target: /
```

5. SEGUIMIENTO Y CONTROL

Finalmente, las horas totales para cada una de las tareas han sido las siguientes:

Nombre de la tarea	Duración horas	
	Previstas	Reales
Preparación	20	20
Análisis	50	52
Documentación	55	48
Diseño	40	40
Desarrollo	170	172
Pruebas	20	30
Seguimiento	14	14
Total	369	376

Figura: Horas reales para cada tarea a realizar

Con lo visto anteriormente y a pesar de las horas invertidas inicialmente en el aprendizaje del framework, se puede afirmar que, a la finalización del TFM, el grado de consecución de objetivos es alto y que se ha logrado ajustar al desarrollo de las horas planificadas para el TFM. Por lo tanto, se puede concluir que el proyecto ha finalizado con éxito.

6. CONCLUSIÓN

Se ha realizado una aplicación de reingeniería. Para ello se llevó a cabo un pequeño estudio con entrevistas a los directores de TFG del Grado de Ingeniería Informática, y un cuestionario que rellenaron los exalumnos. Con esos datos estadísticos, se desarrolló el alcance que ha sido conseguido a excepción de la publicación en el grupo de Facebook debido a cambios en su plataforma. A excepción de ese punto, se ha conseguido una aplicación que reúne todas las características que se propusieron durante la fase de análisis.

Se han actualizado los conocimientos propios sobre PhpStorm, Symfony, Doctrine, Sonata, Bootstrap...

Se han afianzado los conocimientos propios sobre las herramientas JQuery, Charts.js.

Se han superado todos los problemas iniciales con el framework Symfony y el ORM Doctrine, a pesar de los bloqueos continuos al comienzo del proyecto.

Superación del TFM a pesar de todos los contratiempos surgidos durante la realización del mismo.

6.1 Lecciones aprendidas

Una vez concluido el TFM y reflexionando sobre el proceso que se ha llevado a cabo, se refuerza la importancia de la reingeniería de un proyecto. Es importante antes de hacer una mejora estudiar la aplicación y realizar un estudio sobre las necesidades. De este modo, se tiene una garantía de que el producto final se aproximará a lo que busca el usuario.

En este caso, ya había una aplicación informática en uso. Después de realizar un pequeño estudio sobre la misma y de llevar a cabo una serie de entrevistas a los tutores de TFG, así como lanzar un cuestionario para antiguos alumnos, se consiguió reunir una serie de datos que ayudaron a recopilar las necesidades sugeridas por los usuarios de la aplicación.

Por otro lado, se ha podido comprobar que la curva de aprendizaje de los framework y orms puede resultar más pronunciada de lo que se publicita; pero que una vez superada, pueden facilitar el desarrollo de un proyecto. Ciertamente es que nunca se sabe lo que se necesita hasta que no se necesita. Y esto puede suponer un problema en la planificación, ya que no se podrá saber el tiempo necesario para superar el problema surgido.

Hay que mencionar que el uso del framework obliga a utilizar buenas prácticas en la programación. La nomenclatura y la organización redundarán en un mejor código y la estructura de futuros proyectos en Symfony o en otro framework.

6.2 Conclusiones de valoración personal

Mediante la realización del TFM, he podido cubrir los créditos pendientes para la finalización del máster, todo esto siempre y cuando se valore de manera positiva dicho TFM.

A pesar del bloqueo inicial y de tener que abortar y tirar código debido a malas elecciones con el framework y el orm, lo que casi lleva al fracaso de este TFM, se puede decir a día de hoy que todas estas dificultades han sido superadas y se ha conseguido desarrollar una aplicación. Esto me ha producido una gran satisfacción, además de superar momentos duros en el transcurso del TFM, así como otras obligaciones que me han llevado a dejarlo en un segundo plano. Sin embargo, por fin puedo decir que este TFM está concluido.

7. REFERENCIAS BIBLIOGRÁFICAS

Blanco, N. (s.a.) Curso de creación de API en Symfony 3 recuperado de:
<https://openwebinars.net/academia/portada/symfony-api/>

Bootstrap. (s.f.). Recuperado de: <https://getbootstrap.com/>

Bootstrap 4 Tutorial. (s.f.). Recuperado de:
<https://www.w3schools.com/bootstrap4/>

Cancelas, A. (s.a.) Curso de Symfony 3 recuperado de:
<https://openwebinars.net/academia/portada/curso-de-symfony-3/>

Chart.js | Open source HTML5 Charts for your website. (s.f.). Recuperado de:
<https://www.chartjs.org/>

Díaz, C. y Pérez, E. (2017) *Symfony. Desarrollo Web en entorno Servidor* Garceta
ISBN 978-84-1622-894-2.

Fernández, G. (2019, 21 febrero). 20 trucos para Symfony 4. Recuperado de
<https://medium.com/@ger86/20-trucos-para-symfony-4-493f309c0864>

Pacheco, N. (2011) Doctrine 2 ORM Documentation Release 2.1 recuperado de:
https://issuu.com/migueleduardocarmonalugo/docs/manual_doctrine_completo_espanol

Pro Sidebar Template with Bootstrap 4. (s.f.). Recuperado de:
<https://bootsnipp.com/snippets/Q0dAX>

Robles, V. (s.a.) Curso de Symfony 3. Domina el framework PHP más completo
recuperado de: <https://www.udemy.com/curso-de-symfony3-domina-el-framework-php-mas-completo/>

Romero, J. A. (2016, 29 septiembre). Symfony: EntityType para formularios con
relaciones | JAROMERO. Recuperado de: <https://jaromero.es/symfony-entitytype-formularios/>

Symfony, High Performance PHP Framework for Web Development", (s.f.)
Recuperado de: <https://symfony.com>

8. AGRADECIMIENTOS

A los profesores del Departamento de Matemáticas y Computación que me han dedicado unos minutos para realizar el Estudio de la Aplicación.

A los exalumnos del Grado de Ingeniería Informática que han rellenado el cuestionario para obtener información relevante sobre la aplicación.

A mi mujer, a mi hijo y a toda mi familia, gracias a quienes soy quien soy y hacia quienes solo puedo expresar mi sincero agradecimiento por apoyarme durante la etapa académica que hoy culmina.

A los profesores Francisco José y Juan José por haber sido mis guías y mi estímulo durante la redacción de mi TFM.

Anexo I: Plantilla para la entrevista con los directores de TFG

1. ¿Utilizo la aplicación de seguimiento para TFG?
2. En caso de contestar que no la utilizó o la utilizó poco, cuál fue el motivo, qué requisitos le pediría a la aplicación para utilizarla en próximos cursos.
3. En caso de contestar afirmativamente, qué funcionalidad se podría añadir que facilitaría el seguimiento tanto por parte de los alumnos como de los tutores.
4. ¿Qué defectos y virtudes encuentra en la aplicación?
5. Valore los siguientes puntos de 0 a 10 en su utilidad en su TFG siendo 10 la nota más alta y 0 la más baja.
 - A. Módulo plan de seguimiento.
 - B. Módulo de progreso.
 - C. Módulo de evolución del proyecto.
 - D. Módulo búsquedas de otros proyectos.
 - E. Módulo de entregables.
 - F. Módulo de actas.
 - G. Grupo de Facebook.
 - H. La aplicación le ha sido útil.
 - I. Usabilidad de la aplicación.
 - J. Diseño Web.
 - K. Funcionalidad de la aplicación.
 - L. Facilidad de uso.
 - M. Valoración general.
6. Cómo mejoraría la aplicación en aquellos puntos que haya valorado con menos de un seis en la pregunta anterior.
7. Qué le parece el grupo de Facebook, si utilizó este grupo en su elaboración del TFG.
8. ¿Qué le parece tener una red social para comunicarse con otros alumnos de Ingeniería Informática?
9. En caso de que la respuesta anterior sea afirmativa, qué red social aconsejaría.

10. Qué mejoras se podría implementar en la aplicación para atraer futuros alumnos del TFG.
11. Desde su punto de vista, qué le parece que la aplicación dispusiera de las siguientes funcionalidades, valórelas de 0 a 10 siendo 10 la nota más alta.
 - A. Consulta de proyectos de otros compañeros (previa autorización).
 - B. Lecciones aprendidas (valoración una vez acabado el proyecto sobre aquellos problemas relevantes y cuál ha sido la solución).
 - C. Foro de dudas.
 - D. Aviso al correo electrónico de diversos temas mediante un perfil auto configurable.
12. Qué le parece el funcionamiento del programa encargado para subir actas o entregables.
13. Qué le parece que la aplicación guarde los documentos y a petición del tutor le llegue una notificación a su correo avisándole de la entrega, así como del documento.
14. Qué opina de disponer en la aplicación de la siguiente funcionalidad: poder enviar un mensaje a una red social, o por correo a uno o varios alumnos tutorizados.
15. Piensa que esta aplicación puede interactuar con otra aplicación que actualmente se utilice en la UR, y facilite su labor como tutor de alumnos en el TFG.
16. Otras observaciones en relación a la aplicación existente (nuevas funcionalidades, funcionalidades mal diseñadas, funcionalidades que no sirven ...).

Anexo II: Plantilla realizada para el cuestionario de exalumnos

Cuestionario ANÓNIMO con la finalidad de recabar datos que sirvan para la mejora de la aplicación de seguimiento de trabajos fin de grado.

¡Hola !... Soy Álvaro José García Pérez, alumno del Máster en Tecnologías Informáticas. Mi Trabajo Fin de Máster estará basado en una aplicación para ayudar a los estudiantes en el seguimiento de sus trabajos fin de grado, así como facilitar la interacción con sus tutores vía telemática para, de esta manera, optimizar el número de reuniones.

La finalidad de este cuestionario es la de encontrar aspectos a mejorar en la aplicación actual, así como la funcionalidad que actualmente no ofrece.

TIEMPO ESTIMADO PARA REALIZAR EL CUESTIONARIO (10 minutos)

FECHA LÍMITE PARA LA ACEPTACIÓN DE RESPUESTAS: 30 de abril de 2018

1. Año en el que realizaste el Proyecto Fin de Grado
2. Al comenzar el Trabajo Fin de Grado, los tutores os pidieron que os registraraís y utilizaseis la aplicación. Algunos de vosotros os registrasteis y empezasteis a utilizar la aplicación, para posteriormente dejar de utilizarla, ¿cuáles fueron las razones principales para dejar de utilizar la aplicación? Marcar las opciones con las que os identificáis. (No marques ninguna opción si usaste la aplicación hasta el final).
 - A. El profesor no me recomendó utilizarla.
 - B. Me ha costado mucho utilizar la aplicación. No es intuitiva y es difícil llegar a los sitios.
 - C. La estructura del menú no era la apropiada.
 - D. Mi tutor me recomendó utilizar el correo electrónico para la gestión de los entregables.
 - E. Me pareció mejor opción utilizar el correo electrónico para enviar los entregables.

- F. He tenido problemas con la gestión de entregables y esto ha supuesto que deje de utilizarla.
 - G. Las gráficas pueden ser interesantes, pero pierdo tiempo al entrar a la aplicación para buscarlas.
 - H. Las gráficas comparativas con otros alumnos no son reales ya que los datos pueden estar desfasados.
 - I. No me aporta nada la aplicación.
 - J. Otro:
3. Cómo mejorarías la aplicación en aquellos puntos que haya marcado en la pregunta anterior.
- A. El profesor no me recomendó utilizarla.
 - B. Me ha costado mucho utilizar la aplicación. No es intuitiva y es difícil llegar a los sitios.
 - C. El formato de menú no era el apropiado.
 - D. Me pareció mejor opción utilizar el correo electrónico para enviar los entregables.
 - E. He tenido problemas con la gestión de entregables y esto ha supuesto que deje de utilizarla.
 - F. Las gráficas pueden ser interesantes, pero pierdo tiempo al entrar a la aplicación para buscarlas.
 - G. Las gráficas comparativas con otros alumnos no son reales ya que los datos pueden estar desfasados.
 - H. No me aporta nada la aplicación.
 - I. Mi tutor me recomendó utilizar el correo electrónico para la gestión de los entregables.
 - J. Otros. Di cuáles.
4. Qué mejoras se podrían implementar en la aplicación para dar un mejor servicio a futuros alumnos del TFG.
5. Desde tu punto de vista, ¿qué te parece que la aplicación dispusiera de las siguientes funcionalidades? Valóralas de 0 a 10, siendo 10 la nota más alta.

- Conocer los proyectos de perfil similar al mío que se están realizando (título TFG y datos de contacto del alumno y tutor).
- Consulta de proyectos de otros compañeros que han sido depositados en la biblioteca y han otorgado permiso de consulta.
- Lecciones aprendidas (introducir lecciones aprendidas en nuestro proyecto y consultar lecciones aprendidas de otros compañeros).
- Aviso al correo electrónico de novedades en la aplicación (subida de documentos con los que estemos relacionados, por ejemplo, correcciones de nuestro tutor, introducción de lecciones aprendidas que tengan que ver con nuestro proyecto...), mediante un perfil auto configurable.
- Que la aplicación envíe información relativa a fechas de depósitos de trabajos, adjudicación de tribunales, documentos relativos al TFG.
- Que cuando el tutor realice alguna acción sobre tu TFG, te mande un aviso, así como un enlace para acceder directamente.
- Que la aplicación envíe un mensaje, informando del proceso de tu TFG, avisos importantes como, por ejemplo, cuando se aproximen las fechas de depósitos de TFG, asignación de tribunales...
- Que la aplicación tuviera una guía de elaboración de documentos (resumen, introducción, conclusiones...), así como de buenas prácticas al elaborar los diferentes documentos que componen un TFG.
- Que la aplicación disponga de un formulario para rellenar actas de reunión, así como la posibilidad de crear un anexo con todas las actas realizadas durante el TFG, o un anexo para poder consultar todas las actas.
- Mejorar el repositorio de documentos para permitir mantener hilos de versiones de documentos, además de incorporar feedback (entre el tutor y el alumno).
- Que la aplicación proporcione lecciones aprendidas que pueda ayudar a los nuevos alumnos, y que puedas anexas tus lecciones aprendidas a tu memoria.
- Que la aplicación cuente con una parte en el menú con las últimas acciones realizadas.
- Descargar el historial completo del TFG al acabar.

6. Si tu tutor valorase positivamente en su evaluación del TFG la interacción con la aplicación por parte del alumno, ¿habrías utilizado la aplicación?
7. ¿Qué te parece que el tribunal pueda ver cuál ha sido el seguimiento registrado por la aplicación de tu Trabajo Fin de Grado y eso le pueda servir como criterio adicional durante la calificación final del trabajo?
8. ¿Qué te parece que algunas partes del TFG (actas, lecciones aprendidas...) las des de alta en la aplicación y puedas anexarlas a la memoria o anexos de tu TFG mediante una dirección URL?
9. ¿Qué te parece que exista un grupo en una red social donde se encuentren todos los alumnos que han cursado el Grado en Ingeniería Informática?
10. ¿Te parece interesante este grupo para lanzar noticias genéricas (avisos de hitos en la realización TFG, ponencias, talleres...)?
11. En caso de que la respuesta anterior sea afirmativa, ¿qué red social aconsejarías?

Anexo III: Instalación y puesta en marcha del entorno de trabajo

Xampp

El primer paso es instalar xampp, se puede descargar de la siguiente dirección web: <https://www.apachefriends.org/es/index.html> Se deberá elegir una versión con php 7.3.

Composer

Lo primero que se debe hacer es descargar composer e instalarlo. Actualmente, se puede descargar e instalar con los siguientes comandos:

```
php -r "copy('https://getcomposer.org/installer',  
'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===  
'48e3236262b34d30969dca3c37281b3b4bbe3221bda826ac6a9a62d644  
4cdb0dcd0615698a5cbe587c3f0fe57a54d8f5') { echo 'Installer  
verified'; } else { echo 'Installer corrupt';  
unlink('composer-setup.php'); } echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

Cuando se comienza un proyecto es importante decidir la versión de Symfony que se va a utilizar. En Symfony, se puede elegir entre dos versiones actuales: la última estable o la versión LTS (esta versión tiene un mantenimiento de 4 años desde la fecha de lanzamiento). Se pueden consultar las versiones de Symfony en: <https://symfony.com/roadmap>

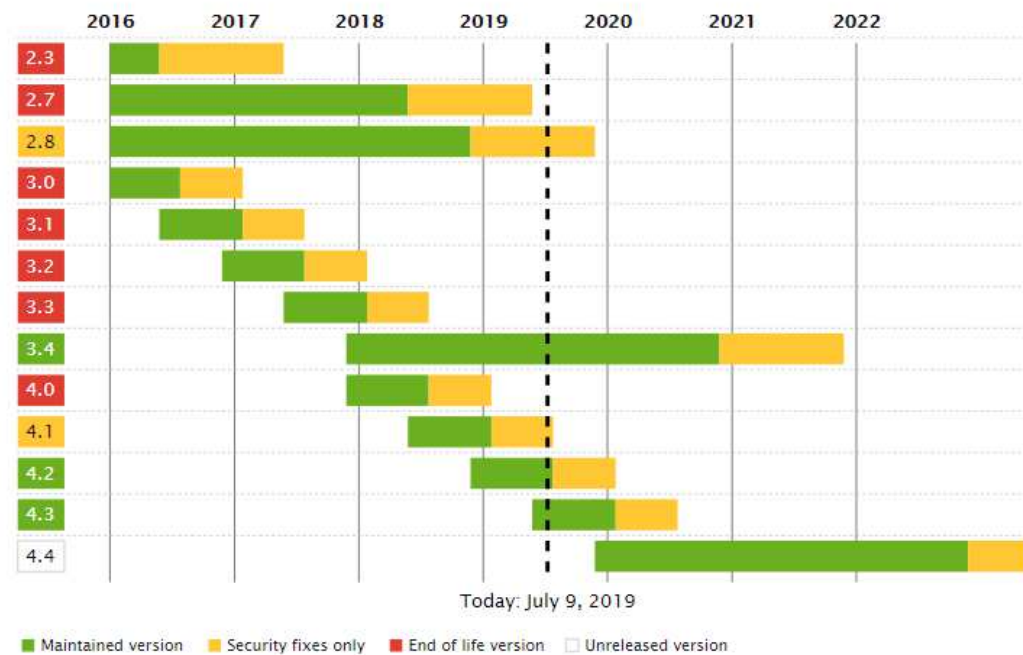


Figura: Versiones de Symfony actualmente

Profiler

Es un componente de Symfony que recoge información de cada petición que recibe la aplicación y la almacena para su análisis posterior. Se instala con el siguiente comando:

```
composer require profiler --dev
```

Instalar el proyecto.

```
symfony new --full --version=3.4 repositario
```

Configuración archive config.inc.php

Al final del archivo se añaden las siguientes líneas:

```
/*Tiempo de ejecución */
```

```
$cfg['ExecTimeLimit'] = 0;
```

Configuración archive php.ini

```
[opcache]
```


; Determines if Zend OPcache is enabled

Zend_extension="php_opcache.dll"

opcache.enable=1

post_max_size=50M

realpath_cache_size=10M

upload_max_filesize=3000M

max_execution_time = 5000

max_input_time = 5000

memory_limit = 6024M

Crear base de datos.

php app/console doctrine:database:create

Crear Virtual Host

Habilitar en httpd.conf la línea

Virtual hosts

Include conf/extra/httpd-vhosts.conf

En el archive httpd-vhost.conf, se creará el virtual host.

<VirtualHost *:80>

DocumentRoot "C:\xampp1\htdocs\repositorio\web"

ServerName repositorio.com

ServerAlias www.repositorio.com

<Directory "C:\xampp1\htdocs\repositorio\web">

Options Indexes FollowSymLinks

AllowOverride All

Order Deny,Allow

Allow from all

</Directory>

</VirtualHost>

Sonata

Se seguirá el siguiente manual para realizar su instalación:

https://symfony.com/doc/current/bundles/SonataAdminBundle/getting_started/installation.html

Si da algún problema de memoria, se ejecutará el siguiente comando:

Php -d memory_limit=-1 c:\composer\composer.phar update --verbose --profile

Para activar el bundle, modificamos el archivo AppKernel.php

```
new Sonata\CoreBundle\SonataCoreBundle(),
```

```
new Sonata\BlockBundle\SonataBlockBundle(),
```

```
new Knp\Bundle\MenuBundle\KnpMenuBundle(),
```

```
new Sonata\DoctrineORMAdminBundle\SonataDoctrineORMAdminBundle(),
```

```
new Sonata\AdminBundle\SonataAdminBundle(),
```

En el archivo config.yml

framework:

```
translator: { fallbacks: ['%locale%'] }
```

PHPStorm

Como IDE de desarrollo se ha empleado PHPStorm, el cual puede descargarse desde: <https://www.jetbrains.com/phpstorm/>

Tareas programadas

Se tiene que crear una tarea programada que se ejecute todos los días con el siguiente comando:

```
curl "http://127.0.0.1:8000/script/enviarCorreos"
```

Anexo IV: Plantilla de ayuda para el desarrollo de Symfony

Arrancar y parar el servidor

php bin/console server:run

php bin/console server:run 0.0.0.0:8000

php bin/console server:stop

<http://localhost:8000/config.php> => para comprobar parámetros

Actualizar vendors

composer update

Caché

php bin/console cache:clear

php bin/console cache:clear --no-warmup

php bin/console cache:clear --env=dev

Assets

php bin/console assets:install

php bin/console assets:install --symlink

php bin/console assets:install --symlink --relative

Esquemas

php bin/console doctrine:generate:entity

php bin/console doctrine:generate:entities AppBundle

php bin/console make:entity --regenerate

php bin/console doctrine:schema:update --dump-sql

php bin/console doctrine:schema:update --force

```
php bin/console doctrine:generate:crud
```

```
php bin/console doctrine:generate:form AppBundle:Persona
```

```
php bin/console sonata:admin:generate AppBundle/Entity/Director
```

```
php bin/console doctrine:generate:form AppBundle:Persona
```

```
php bin/console security:encode-password
```

Comprobar debilidades

```
php bin/console security:check
```

Doctrine

Ejemplo de anotaciones.

```
/**
 * Area
 *
 * @ORM\Table(name="area")
 *
 * @ORM\Table(name="area", uniqueConstraints={@ORM\UniqueConstraint(name="unq_area_nombre", columns={"nombre"})}, indexes={@ORM\Index(name="idx_area", columns={"nombre"})})
 * @ORM\Entity(repositoryClass="AppBundle\Repository\AreaRepository")
 */
class Area{
```

Relaciones uno a uno, bidireccionales

```
class Persona implements UserInterface{
    /**
     * @var int
     *
     * @ORM\Column(name="idPersona", type="integer")
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    private $idPersona;
    /**
     * @var Alumno
     * One Customer has One Alumno
     * @ORM\OneToOne(targetEntity="Alumno", mappedBy="persona")
     */
    private $alumno;
}

class Alumno{
    /**
     * @var int
     * @ORM\Id
     * @ORM\Column(name="idAlumno", type="integer", nullable=true)
     */
    private $idAlumno;
}
```

```
/**
 * One Persona has One Persona.
 * @var Persona
 * @ORM\OneToOne(targetEntity="Persona", inversedBy="alumno")
 * @ORM\JoinColumn(name="idAlumno", referencedColumnName="idPersona")
 */
private $persona;
```

Relaciones uno a muchos, bidireccionales

```
class Area{
    ##
    /**

    * @ORM\OneToMany(targetEntity="Tfg", mappedBy="area")
    */
    protected $tfgs;
    public function __construct() {
        $this->tfgs = new ArrayCollection();
    }

class Tfg{
    /**
    * @var int
    *
    * @ORM\Column(name="idArea", type="integer", nullable=true)
    */
    private $idArea;
    /**
    * @var Area
    * @ORM\ManyToOne(targetEntity="Area", inversedBy="tfgs")
    * @ORM\JoinColumn(name="idArea", referencedColumnName="idArea")
    */
    private $area;
```

Relaciones muchos a muchos, bidireccionales.

```
class Director{
    /**
    * @var int
    * @ORM\Id
    * @ORM\Column(name="idDirector", type="integer")
    */
    private $idDirector;
    /**
    * @ORM\ManyToMany(targetEntity="Tfg", mappedBy="directores")
    */
    protected $tfgs;
    public function __construct()
    {
        $this->tfgs = new ArrayCollection();
    }

class Tfg{
    /**
    * @ORM\ManyToMany(targetEntity="Director", inversedBy="tfgs")
    * @ORM\JoinTable(name="tfgs_directores",
    *                 joinColumns={@ORM\JoinColumn(name="idTfg",
    referencedColumnName="idTfg")},
    *                 inverseJoinColumns={@ORM\JoinColumn(name="idDirector",
    referencedColumnName="idDirector")})
    */
```

```

*/
protected $directores;
public function __construct()
{
    $this->directores = new ArrayCollection();
}

```

Métodos repository

```

public function findByCodigoDescarga($codigo)
{
    $query = $this->createQueryBuilder('f')
        ->select('f')
        ->leftJoin('f.subida', 's')
        ->leftJoin('s.persona', 'sp')
        ->where('s.codigoDescarga= :codigoDescarga')
        ->setParameter('codigoDescarga', $codigo)
        ->orderBy('f.fecha', 'asc')
        ->getQuery();
    return $query->getResult();
}

```

```

public function devolverDatosEstadisticos($dias) {

```

```

    $csql="select ROUND(avg(horasEmpleadas),0) as n1,
                ROUND(avg(horasPlanificadas),0) as n2,
                ROUND(avg(porPlanificacion),0) as n3,
                ROUND(avg(porAnalisis),0) as n4,
                ROUND(avg(porDiseno),0) as n5,
                ROUND(avg(porProducto),0) as n6,
                ROUND(avg(horasPlanificacion),0) as n7,
                ROUND(avg(horasAnalisis),0) as n8,
                ROUND(avg(horasDiseno),0) as n9,
                ROUND(avg(horasProducto),0) as n10,
                COUNT(horasEmpleadas) as n11
    from Progreso
    where ultimo=true and ((porPlanificacion=100 and porAnalisis=100
and porDiseno=100 and porProducto=100) or
    (fechaActualizacion>= date_sub(now(), INTERVAL ".$dias." day)));
    ";
    $db=$this->_em->getConnection();
    $stmt=$db->prepare($csql);
    $params=array();
    $stmt->execute($params);
    $result=$stmt->fetchAll();
    $resultado=($result[0]);

    $progreso=new Progreso();
    $progreso->setHorasEmpleadas($resultado['n1']);
    $progreso->setHorasPlanificadas($resultado['n2']);
    $progreso->setPorPlanificacion($resultado['n3']);
    $progreso->setPorAnalisis($resultado['n4']);
    $progreso->setPorDiseno($resultado['n5']);
    $progreso->setPorProducto($resultado['n6']);
    $progreso->setHorasPlanificacion($resultado['n7']);
    $progreso->setHorasAnalisis($resultado['n8']);
    $progreso->setHorasDiseno($resultado['n9']);
    $progreso->setHorasProducto($resultado['n10']);
    $progreso->setNumeroSeguimiento($resultado['n11']);
    return $progreso;
}

```

Acceder a la sesión

```
$persona=$this->getUser();
$alumno=($this->get('session')->get('alumno'));

$this->get('session')->getFlashBag()->add('danger', 'Usuario no logueado');

{{app.user}}
{{app.session.get('alumno')}}
```

Recoger parámetros get

```
$codigo=($request->query->get("codigo"));
```

Formularios

```
->add('dni', null, array(
    'label' => 'DNI',
    'label_attr' => array('class' => 'col-sm-2 col-form-label text-right'),
    'attr' => array('class' => 'form-control', 'placeholder' => 'DNI')
))

->add('tipo', EntityType::class, array(
    'label' => 'Tipo',
    'class' => 'AppBundle:Tipo',
    'choice_label' => 'nombre',
    'label_attr' => array('class' => 'col-sm-2 col-form-label text-right'),
    'attr' => array('class' => 'form-control', 'placeholder' => 'Tipo'))

{{ form_start(form, {'attr': {'id':form.vars.id,'onsubmit':'return
validarFormulario(this);','action':''} } ) }}

<div class=" form-group row ">
    {{form_label(form.avisosPropios)}}
    <div class="col-sm-1">
        {{ form_widget(form.avisosPropios)}}
        {{ form_errors(form.avisosPropios) }}
    </div>
</div>
```

Control de errores

```
{% for label, flashes in app.session.flashbag.all %}
    {% for flash in flashes %}
        <div class="alert alert-{{label}} alert-dismissible">
            <button type="button" class="close" data-
dismiss="alert">&times;</button>
            {{ flash }}
        </div>
    {% endfor %}
{% endfor %}
```


Anexo V: Casos de uso

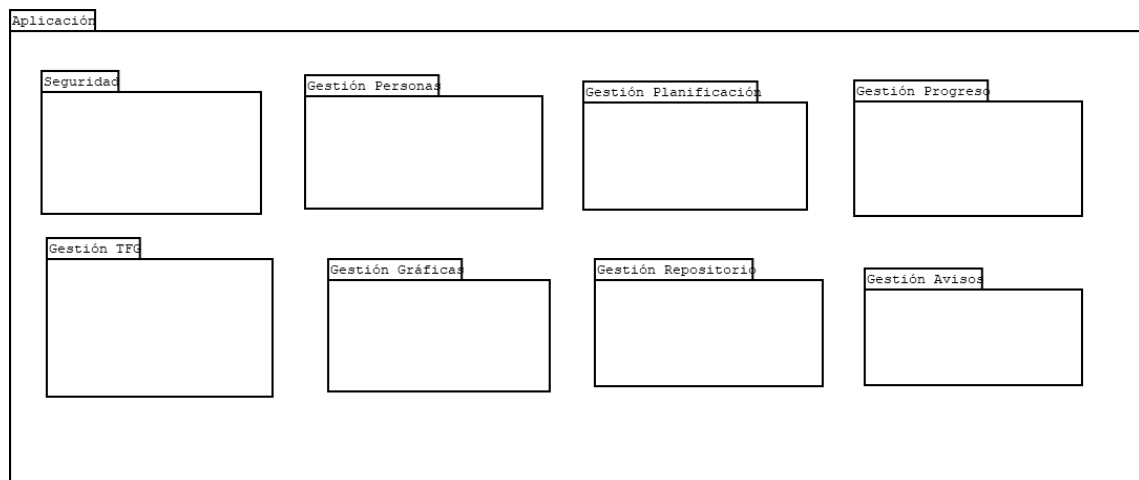


Figura: Diagrama de paquetes - Aplicación

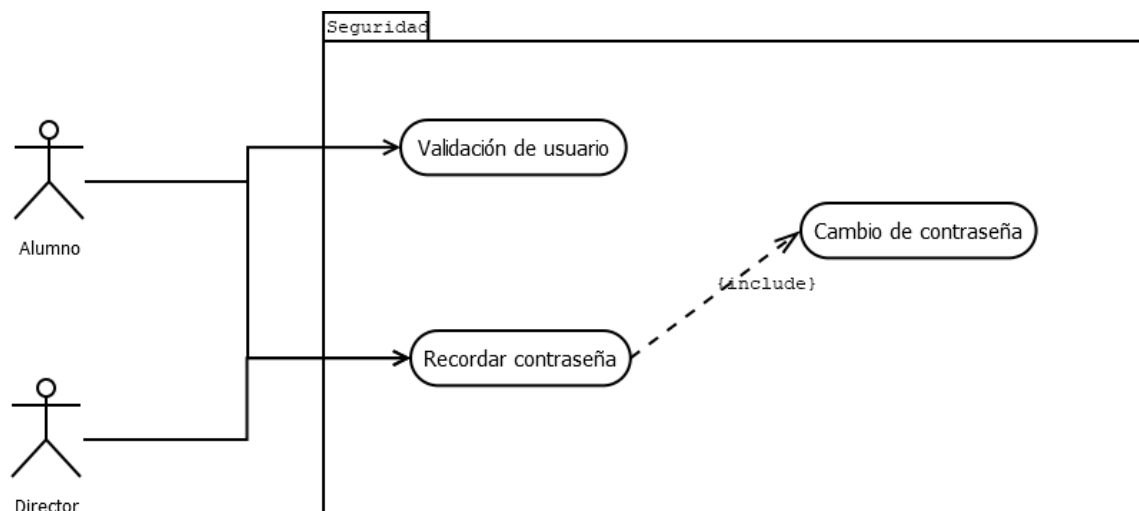


Figura: Diagrama de casos de uso - Seguridad

C.U 001 - Validación de usuario

Definición

El usuario del sistema se identifica y autentifica para poder acceder al sistema. Solamente accederán al sistema aquellos usuarios registrados, con excepción de alta alumno, alta director (sin requisitos), descarga de documentos, ver

	feedback de documentos (a través de un token de seguridad).	
Requisitos relacionados	S1	
Precondición	El usuario debe estar dado de alta en el sistema.	
Postcondición	Autenticado el usuario se le asignará un perfil.	
Observación		
Secuencia de pasos	Paso	Acción
	1	Se inicia cuando el usuario intenta acceder a la aplicación. Mostrará formulario de usuario y contraseña.
	2	El usuario introduce datos y envía el formulario.
	3	El sistema valida los datos contra la BD y le asigna un perfil.
	4	Se le muestran las opciones a las que tiene acceso el usuario.
Secuencia de excepciones	Paso	Excepción
	3	Si el usuario no es válido, se le devolverá al formulario de identificación.

C.U 002 – Recordar contraseña		
Definición	Generación de una nueva contraseña, en caso de olvido de la misma, o posibilidad de sustracción.	
Requisitos relacionados	S1, S2.	
Precondición	El usuario debe estar dado de alta previamente.	
Postcondición	La contraseña debe tener entre 6 y 12 caracteres, en los que se deben incluir tres tipos de los siguientes: mayúsculas, minúsculas, números, caracteres especiales.	
Observación	El token de seguridad tendrá una vigencia de 24 horas a excepción de haberse utilizado anteriormente.	
	Paso	Acción

Secuencia de pasos	1	Se introducirá el correo electrónico y DNI mediante un formulario.
	2	Se comprueba la existencia en la BD de un usuario con ese correo electrónico y DNI. Se genera un token que se envía por correo electrónico.
	3	Se introducirá el token recibido en el correo electrónico. Se comprobará que existe dicho token y se mostrará un formulario.
	4	El usuario introducirá su nueva contraseña la cual deberá tener entre 6 y 12 caracteres además de incluir tres tipos de los siguientes: mayúsculas, minúsculas, números, caracteres especiales.
Secuencia de excepciones	Paso	Excepción
	2	En caso de que no exista relación entre correo electrónico y DNI, se mostrará un error. En caso de que no se pueda enviar el correo electrónico, se mostrará un error.
	3	En caso de que el token haya sido utilizado o transcurridas 24 horas desde su creación se notificará que el token ya no es válido.
	4	Si el usuario no introduce la misma contraseña o no tiene los requisitos solicitados, se mostrará un mensaje de error.

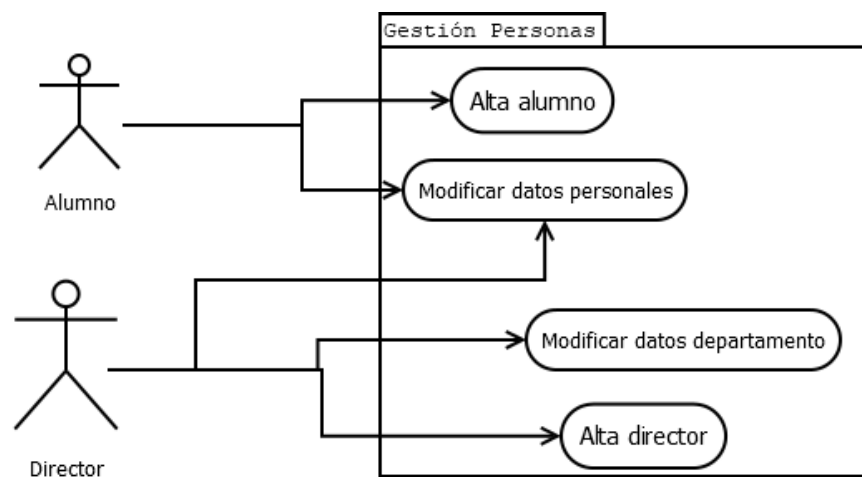


Figura: Diagrama de casos de uso - Gestión de personas

C.U 003 – Alta alumno		
Definición	Se dará de alta un nuevo alumno.	
Requisitos relacionados	P1, P2, P3.	
Precondición	No puede estar dado de alta previamente; se considerará si coincide nombre, apellido1, apellido2 y DNI.	
Postcondición	<p>No podrá haber más de una persona con el mismo correo.</p> <p>No podrá haber más de una persona con el mismo identificador de usuario.</p> <p>Se debe introducir la misma contraseña y esta debe reunir los criterios enunciados anteriormente.</p>	
Observación		
Secuencia de pasos	Paso	Acción
	1	Formulario con los datos genéricos del alumno.
	2	Inserción en la BD.
	3	Se crean en la BD todas las clases involucradas: persona, alumno, avisos.
Secuencia de excepciones	Paso	Excepción
	3	En caso de que la persona, el correo electrónico o el identificador de usuario ya existieran, se notificará el error.

C.U 004 – Alta Director	
Definición	Se dará de alta un nuevo director.
Requisitos relacionados	P1, P2, P3.

Precondición	No puede estar dado de alta previamente; se considerará si coincide nombre, apellido1, apellido2 y DNI.	
Postcondición	<p>No podrá haber más de una persona con el mismo correo.</p> <p>No podrá haber más de una persona con el mismo identificador de usuario.</p> <p>Se debe introducir la misma contraseña y esta debe reunir los criterios enunciados anteriormente.</p>	
Observación		
Secuencia de pasos	Paso	Acción
	1	Formulario con los datos genéricos del director.
	2	Inserción en la BD.
	3	Se crea en la BD todas las clases involucradas: persona, director, avisos.
Secuencia de excepciones	Paso	Excepción
	3	En caso de que la persona, el correo electrónico o el identificador de usuario ya existieran, se notificará el error.

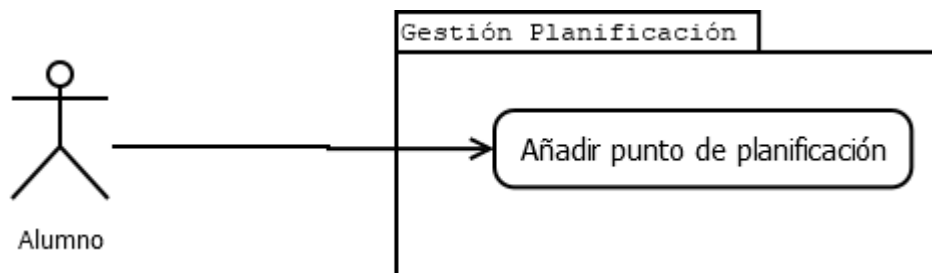


Figura: Diagrama de casos de uso - Gestión de planificación

C.U 005 – Añadir punto de planificación	
Definición	Se añadirán nuevas planificaciones. En el caso de que ese mismo día hubiese insertado una planificación, lo que se realizará será una modificación de la misma.
Requisitos relacionados	S1, L1, L2, L3, L4, L5, L6, L7, L8.

Precondición	Debe haberse dado de alta previamente el TFG, sino no será posible realizar una planificación.	
Postcondición	Las horas en los puntos de control no podrán ser mayores que las horas al concluir el TFG.	
Observación		
Secuencia de pasos	Paso	Acción
	1	Formulario con los datos de la planificación y aquellos puntos de control que hayan pasado de la fecha, serán solo de lectura.
	2	Insertión o actualización en la BD.
	3	Envío de correo electrónico a los directores y al alumno (configuración de avisos).
	4	Redireccionamiento a la página de inicio.
Secuencia de excepciones	Paso	Excepción
	2	Se comprobará que las horas totales planificadas no sean menores que en ningún punto de control.
	3	En caso de que el servicio de correo electrónico esté caído.

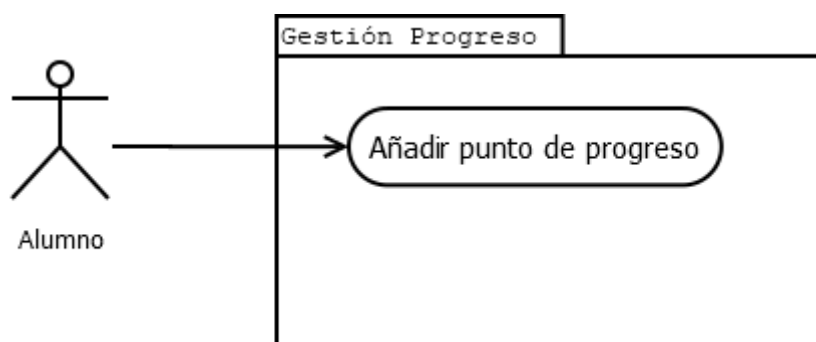


Figura: Diagrama de casos de uso - Gestión de progreso

C.U 006 – Añadir punto de progreso	
Definición	Se añadirán nuevos progresos; en el caso de que ese mismo día hubiese insertado un seguimiento, lo que se realizará será una modificación del mismo.

Requisitos relacionados	S1, D1, D2, D3, D4, D5.	
Precondición	Debe haberse dado de alta previamente una planificación, sino no será posible realizar un seguimiento.	
Postcondición	Las horas totales deberán corresponder a la suma de las horas parciales. El valor de los porcentajes deberá estar entre 0 y 100. Solo se podrá introducir un seguimiento si está entre la fecha de inicio y de fin.	
Observación		
Secuencia de pasos	Paso	Acción
	1	Formulario con los datos del progreso; el alumno los rellenará y enviará.
	2	Inserción en la BD.
	3	Envío de correo electrónico a los directores y al alumno (configuración de avisos).
	4	Redireccionamiento a la página de inicio.
Secuencia de excepciones	Paso	Excepción
	2	En caso de que algún parámetro no coincida con las postcondiciones, saltará una excepción.
	3	En caso de que el servicio de correo electrónico esté caído.

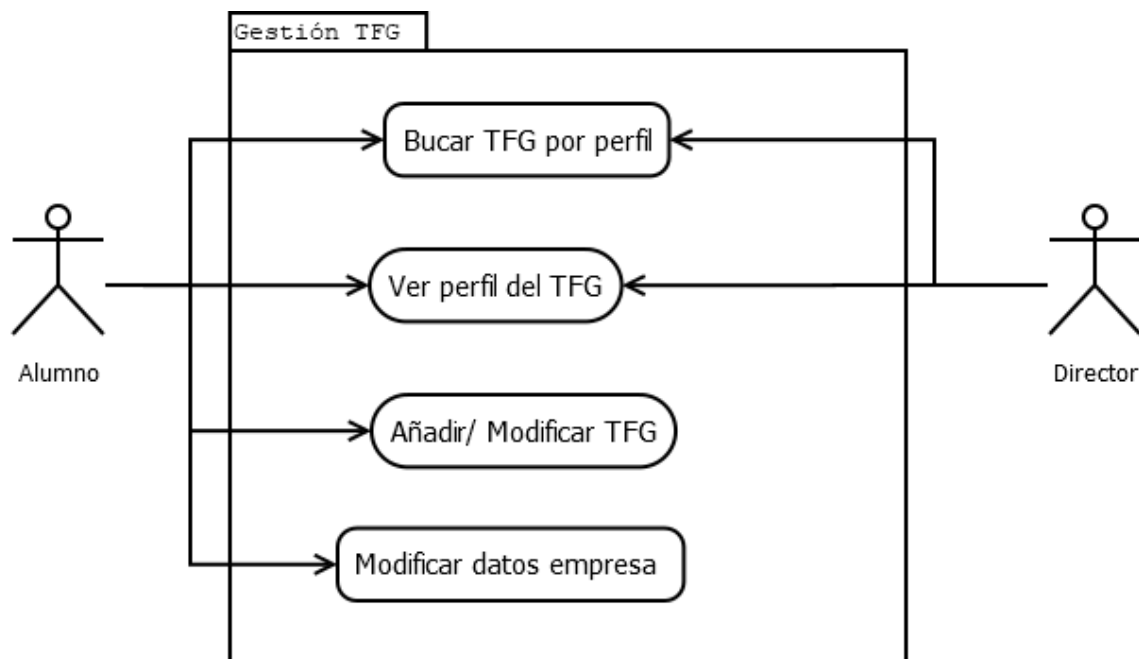


Figura: Diagrama de casos de uso - Gestión de TFGS

C.U 007 – Añadir/Modificar TFG		
Definición	El alumno configura el perfil de su TFG (Título, Tipo de TFG, Área, Tecnologías empleadas), el alumno solo podrá introducir los directores que dirigirán su TFG la primera vez.	
Requisitos relacionados	S1, T1, T2, T3, T4.	
Precondición	El alumno tendrá que estar dado de alta en el sistema.	
Postcondición		
Observación	En caso de que una tecnología no se encuentre en la lista, el alumno podrá añadirla desde el formulario.	
Secuencia de pasos	Paso	Acción
	1	Se representará en la pantalla un formulario con los datos actuales del TFG; en caso de no tener ninguno, aparecerá sin rellenar.
	2	El alumno rellenará el formulario y lo enviará.
	3	Se guardará en la BD en las tablas afectadas.

	4	Se reenviará al alumno a la pantalla de inicio.
Secuencia de excepciones	Paso	Excepción

C.U 008 – Modificar datos empresa

Definición	Se mostrará en la pantalla un formulario con los datos de la empresa donde realiza el alumno su TFG. El alumno podrá modificar los datos de la empresa o borrarlos en caso de que ya no haga su TFG en la empresa.	
Requisitos relacionados	S1, T5, T6.	
Precondición		
Postcondición		
Observación		
Secuencia de pasos	Paso	Acción
	1	Se mostrará en un formulario los datos de la empresa donde realiza el TFG.
	2	El usuario rellenará el formulario que se valida y se comprobará que los campos sean correctos.
	3	Se actualizan los datos en la BD.
Secuencia de excepciones	Paso	Excepción

C.U 009 – Buscar TFG por perfil

Definición	Se rellenan los datos (Área, Tipo de proyecto y Tecnología empleada). Tras introducirlos, se muestran los TFG que coinciden con dichos criterios.	
Requisitos relacionados	S1.	
Precondición		

Postcondición			
Observación			
Secuencia de pasos	Paso	Acción	
	1	Rellenar los criterios de búsqueda.	
	2	Se mostrarán por pantalla todos los datos de los TFG que coincidan con los criterios de búsqueda.	
Secuencia de excepciones	Paso	Excepción	

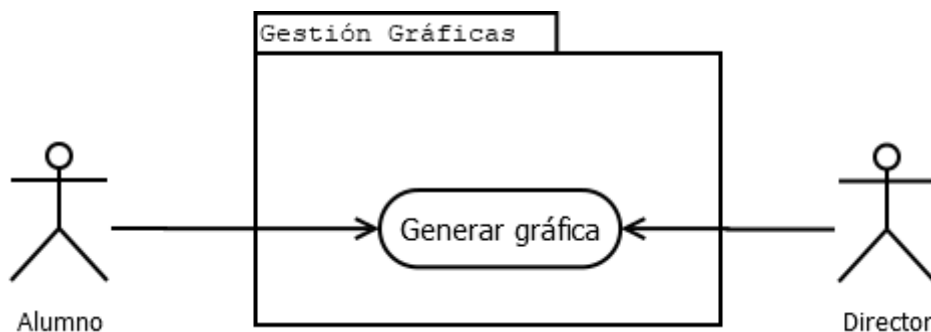


Figura: Diagrama de casos de uso - Gestión de gráficas

C.U 010 – Generar gráfica	
Definición	Se generarán diferentes gráficas con el objetivo de informar de los datos recopilados por la aplicación, tanto del TFG de un alumno como de la media de los diferentes TFG.
Requisitos relacionados	S1, E1, E2, E3, E4, E5, E6, E7, E8.
Precondición	En el caso del director del TFG, tendrá que tener seleccionado un alumno antes de acceder a las estadísticas.
Postcondición	

Observación	Mediante un id, se seleccionará qué tipo de gráfico se quiere mostrar.		
Secuencia de pasos	Paso	Acción	
	1	Se seleccionará desde el menú el tipo de gráfico a mostrar.	
	2	Se pintará con canvas la gráfica solicitada.	
Secuencia de excepciones	Paso	Excepción	

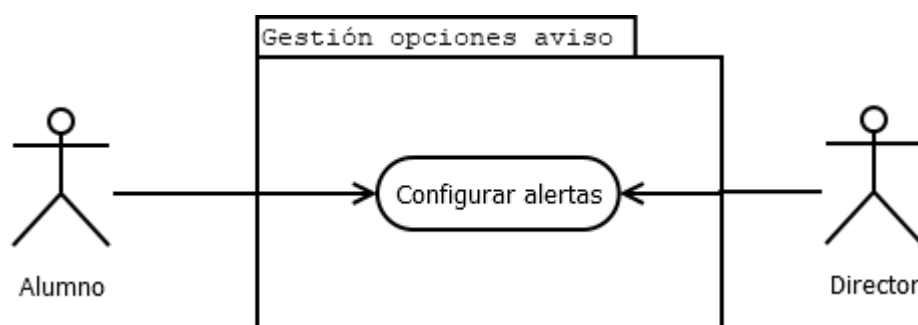


Figura: Diagrama de casos de uso - Gestión de opciones de aviso

C.U 011 – Configurar alertas		
Definición	Se mostrarán los diferentes tipos de aviso, y la persona configurará según sus preferencias.	
Requisitos relacionados	S1, A1, A2.	
Precondición	Se crearán las opciones de aviso, activando todas al dar de alta una persona.	
Postcondición		
Observación	Se recomienda que los alumnos dejen marcadas todas las opciones de aviso, para en el caso de que metan datos erróneos sean conscientes de estos.	
	Paso	Acción

Secuencia de pasos	1	Formulario con las opciones de aviso.
	2	Actualización en la BD de las opciones de aviso.
Secuencia de excepciones	Paso	Excepción

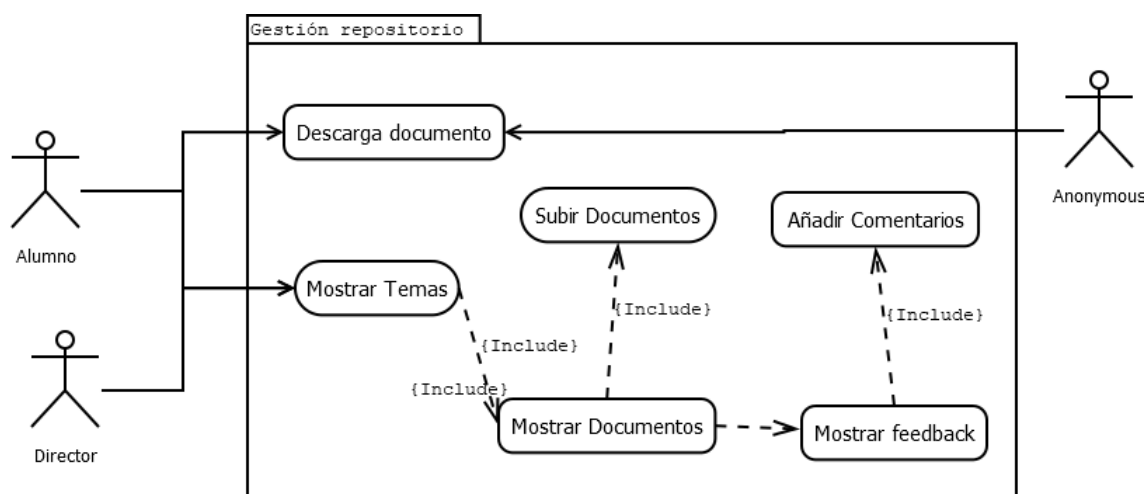


Figura: Diagrama de casos de uso - Gestión de repositorio

C.U 012 – Descarga de documento			
Definición	Cualquier persona que tenga el token correspondiente a un documento, podrá descargar dicho documento.		
Requisitos relacionados	S1, R3.		
Precondición			
Postcondición	.		
Observación			
Secuencia de pasos	Paso	Acción	
	1	Se introducirá mediante url el código token del archivo para su descarga.	
	2	Se descargará el archivo en el dispositivo.	

Secuencia de excepciones	Paso	Excepción
	2	En caso de no existir el token, saltará una excepción.

C.U 013 – Mostrar temas

Definición	Se mostrará en la pantalla con formato de foro los diferentes tipos de documentos, mostrando además el número de archivos subidos, mensajes en ese tema, último documento con modificaciones y el último comentario.		
Requisitos relacionados	S1.		
Precondición			
Postcondición			
Observación	El alumno solo verá los datos generados de su TFG; en cambio, el director verá los datos de un TFG si tiene seleccionado un alumno, y en otro caso verá todos los datos relativos a los alumnos que dirige.		
Secuencia de pasos	Paso	Acción	
	1	Aparece una página web en forma de foro.	
	2	Dependiendo donde pulse, podrá ir a los datos de un tipo de documento, a un documento concreto o descargar un documento.	
Secuencia de excepciones	Paso	Excepción	
	2	En caso de que el documento no exista, se lanzará una excepción.	

C.U 014 – Mostrar documentos

Definición	Se mostrarán en la pantalla los documentos correspondientes a un tema, así como el último comentario de cada uno de dichos documentos.
-------------------	--

Requisitos relacionados	S1, R1, R2, R3, R5.	
Precondición		
Postcondición		
Observación	El alumno solo verá los documentos de su TFG; en cambio, el director verá los documentos de un TFG si tiene seleccionado un alumno, y en otro caso verá todos los documentos relativos a los alumnos que dirige.	
Secuencia de pasos	Paso	Acción
	1	Aparece una página web en forma de foro.
	2	Dependiendo donde pulse podrá ir a los comentarios de un documento, descargar un documento o subir un nuevo documento.
Secuencia de excepciones	Paso	Excepción
	2	En caso de que el documento no exista, se lanzará una excepción.

C.U 015 – Mostrar feedback		
Definición	Se mostrará en la pantalla un documento con todos los feedback recibidos con formato de foro.	
Requisitos relacionados	S1, R3, R4.	
Postcondición	Solo podrán realizar comentarios a un documento los directores del TFG o el alumno implicado.	
Observación		
Secuencia de pasos	Paso	Acción
	1	Aparece una página web en forma de foro.
	2	Dependiendo donde pulse, podrá descargar un documento o realizar un comentario.
Secuencia de excepciones	Paso	Excepción